



MASTERARBEIT

Herr
Alexander Fiedler

Integration eines Workflow- Management-Systems in ein ERP-System für kleine und mittlere Unternehmen

Analyse, Entwurf und prototypische Umsetzung

2016



MASTER THESIS

Mr.
Alexander Fiedler

Integration of a workflow management system into an ERP system for small and medium-sized enterprises

Analysis, design and prototypical implementation

2016

MASTERARBEIT

Integration eines Workflow-Management- Systems in ein ERP-System für kleine und mittlere Unternehmen

Analyse, Entwurf und prototypische Umsetzung

Autor:

Alexander Fiedler

Studiengang:

Informatik

Seminargruppe:

IF10w1-M

Erstprüfer:

Prof. Dr. rer. biol. hum. Rudolf Stübner

Zweitprüfer:

M. Sc. Michael Hertel

Mittweida, 2016

MASTER THESIS

Integration of a workflow management system into an ERP system for small and medium-sized enterprises

Analysis, design and prototypical implementation

Author:

Alexander Fiedler

Study Programme:

Informatics

Seminar Group:

IF10w1-M

First Referee:

Prof. Dr. rer. biol. hum. Rudolf Stübner

Second Referee:

M. Sc. Michael Hertel

Mittweida, 2016

Bibliografische Angaben

Fiedler, Alexander: Integration eines Workflow-Management-Systems in ein ERP-System für kleine und mittlere Unternehmen, Analyse, Entwurf und prototypische Umsetzung, 76 Seiten, 26 Abbildungen, 8 Tabellen, Hochschule Mittweida, University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften

Masterarbeit, 2016

Satz: L^AT_EX

Referat

Standardsoftware wie Enterprise Resource Planning Systeme bieten eine ganzheitliche Unterstützung der Wertschöpfungsprozesse eines Anwenderunternehmens. Die darüber abgebildeten Geschäftsprozesse spiegeln im Allgemeinen das Wissen eines Entwicklerunternehmens über die Vorgehensweisen und Methoden innerhalb einer bestimmten Branche wider. In der heutigen Zeit gehören Enterprise Resource Planning Systeme zur softwaretechnischen Grundausstattung für die überwiegende Mehrzahl Kleiner und Mittlerer Unternehmen. Deren primärer Wettbewerbsvorteil liegt hierbei vor allem in ihrer Individualität, welche durch ein Enterprise Resource Planning System zu unterstützen ist. Diesbezügliche Anpassungen des Systems sind aufgrund seiner Komplexität häufig sehr kostenintensiv und übersteigen nicht selten das Budget eines Kleinen und Mittleren Unternehmens.

Workflow-Management-Systeme bieten eine einfache Möglichkeit die internen Geschäftsprozesse eines Anwenderunternehmens kostengünstig über ein Enterprise Resource Planning Systems abzubilden. Gleichzeitig erlauben sie die Steuerung der modellierten Arbeitsabläufe.

Diese Arbeit beschäftigt sich daher mit der Integration eines Workflow-Management-Systems in ein bestehendes Enterprise Resource Planning System. Das Integrationskonzept bezieht sich dabei ausschließlich auf Client-Server-Architekturen. Auf Basis einer IST-Analyse wird dabei ein Integrationskonzept für eine lose Kopplung beider Systeme entwickelt. Im Ergebnis soll dies einen einfachen Austausch des Workflow-Management-Systems garantieren und dessen Funktionen innerhalb des Enterprise Resource Planning System bereitstellen.

Schlüsselwörter: Workflow Management, Workflow-Management-System, Enterprise Resource Planning, Integration, Integrationskonzept, Kleine und Mittlere Unternehmen

Bibliographic Information

Fiedler, Alexander: Integration of a workflow management system into an ERP system for small and medium-sized enterprises, Analysis, design and prototypical implementation, 76 pages, 26 figures, 8 tables, Hochschule Mittweida, University of Applied Sciences, Faculty of Applied Computer and Life Sciences

Master Thesis, 2016

Typeset by L^AT_EX

Abstract

Standard software like enterprise resource planning systems provide a comprehensive support of the value chain of user companies. The business processes represented by ERP systems typically reveal the knowledge of the development company about the approaches and methods in the specific industrial sectors. In modern times enterprise resource planning systems are part of basic software configuration in the majority of small and medium-sized enterprises. The competitive advantage of these companies is mainly their individuality, which should be supported by their ERP systems. Therefore required modifications of the systems are often cost intensive because of their complexity and thus often exceed the budget of small and medium-sized businesses.

Workflow management systems provide a simple opportunity for modeling the internal business processes of a company affordable by an enterprise resource planning system. Simultaneously they allow to control the modelled workflow.

Therefore this work addresses the integration of a workflow management system in an existing enterprise resource planning system. The integration concept refers only to client server architectures. Based on an actual state analysis the integration concept for a loose coupling of the both systems is developed. The result should guarantee a simple exchange of the workflow management system and provide its functionality in the enterprise resource planning system.

Keywords: workflow management, workflow management system, enterprise resource planning, integration, integration concept, small and medium-sized enterprises

I. Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	II
Tabellenverzeichnis	III
Abkürzungsverzeichnis	IV
1 Einleitung	1
1.1 Motivation	1
1.2 Ziel	2
1.3 Abgrenzung	3
1.4 Aufbau	3
2 Grundlagen	5
2.1 Kleine und mittlere Unternehmen	5
2.1.1 Begriffsdefinition	5
2.1.2 Qualitative Merkmale	6
2.2 ERP-Systeme für KMU	8
2.2.1 Begriffsdefinition	8
2.2.2 Merkmale beziehungsweise Bestandteile heutiger ERP-Systeme	10
2.2.3 Vor- und Nachteile durch den Einsatz eines ERP-Systems	12
2.3 Workflow-Management-Systeme für KMU	13
2.3.1 Grundlegende Begrifflichkeiten	13
2.3.2 Begriffsdefinition	19
2.3.3 Referenzarchitektur der WfMC	20
2.3.4 Architekturkonzepte für WfMS	26
2.4 Integration von Workflow-Management-Systemen	30
2.4.1 Integrationsarten von WfMSe	30
2.4.2 Integrationsstufen für WfMSe	30
3 Beschreibung der Ausgangssituation	34
3.1 ERP-System hscERP	34
3.1.1 Zugrundeliegende Architekturkonzepte	35
3.1.2 Interne Unterstützung der Architekturkonzepte	36
3.1.3 Abbildung der Benutzer- und Rechteverwaltung	37
3.2 Workflow-Management-System jBPM	39
3.2.1 Grundlegendes	39
3.2.2 Komponenten des WfMSs jBPM	39
3.2.3 Komponenten von jBPM unter Berücksichtigung des WfMC-Referenzmodells	42
3.2.4 Workflow-Modellierung und -Ausführung in jBPM	42
3.2.5 Human Task in jBPM	43
3.2.6 Überblick relevante API-Elemente	45

3.3	Anforderungen an das Integrationskonzept	48
3.3.1	Allgemeine Anforderungen	48
3.3.2	Funktionale Anforderungen	49
3.3.3	Nichtfunktionale Anforderungen	50
4	Integrationskonzept für das hscERP	51
4.1	Architektur des Gesamtsystems	51
4.2	Datenbank-Komponente	52
4.2.1	Datenmodell des Gesamtsystems	52
4.2.2	Datenintegritätsprüfung	54
4.3	Server-Komponente	54
4.3.1	hscERP-Komponente	55
4.3.2	WfMS Abstraktionsschicht	55
4.3.3	WfMS Implementierung	56
4.4	Client-Komponente	56
4.4.1	Worklist	57
4.4.2	Workflow-Management-Modul	58
5	Prototypische Implementierung	59
5.1	Realisierung der Datenhaltung	59
5.2	Realisierung der Server-Komponente	59
5.2.1	Implementierung des Verbindungsaufbaus	59
5.2.2	Implementierung des Workflow-Instanz- und User-Task-Zugriffs	61
5.3	Realisierung der Client-Komponente	64
6	Zusammenfassung und Ausblick	65
A	WfMC Metamodell	66
B	WAPI-Funktionen Invoked Applications	67
C	Überblick aller Anwendungsfälle	69
	Literaturverzeichnis	71
	Glossar	74

II. Abbildungsverzeichnis

2.1	Anzahl der KMU in Deutschland nach Beschäftigtenzahl	5
2.2	Workflow-Arten nach Automatisierungsgrad	16
2.3	Beispiel eines Workflow-Modelles	18
2.4	Workflow-Regelkreis	18
2.5	Referenzarchitektur der WfMC	21
2.6	Workflow-Metamodell der WfMC	22
2.7	Interoperabilitätsmodelle für WfMs	25
2.8	Form-based Architektur	26
2.9	Engine-based Architektur	27
2.10	Client-Server-Schichtenmodell	28
3.1	Schematischer Aufbau hscERP	34
3.2	Portierung der Zwei-Schichten-Architektur auf eine Drei-Schichten-Architektur	35
3.3	Interne Kommunikationsbeziehung zwischen Klassen	36
3.4	Übersicht Tabellen für Benutzer- und Rechteverwaltung	38
3.5	Module der Excution-Komponente	41
3.6	Human-Task-Lebenzyklus	44
4.1	Abstrahierte Übersicht auf das Gesamtsystem	51
4.2	Datenbankmodell für die Integration des WfMS	53
4.3	Zugangsbereiche für unterschiedliche Benutzerrollen	56
5.1	Implementierung des grundlegenden Verbindungsaufbaus	60
5.2	Implementierung der Workflow-Instanz- und User-Task-Verwaltung	61
5.3	Implementierung User-Task-Manager	62
5.4	Implementierung Instance-Manager	63
5.5	Implementierung des Client-Zugriffs	64
C.1	Übersicht der Anwendungsfälle für das Workflow-Management-Modul	69
C.2	Übersicht der Anwendungsfälle für die Worklist	70

III. Tabellenverzeichnis

1.1 Verbreitung von ERP-Systemen in Deutschland	1
2.1 Klassifizierung Klein- und Mittelunternehmen der EU	6
2.2 Qualitative Merkmalsunterscheidung zwischen KMU und Großunternehmen	6
2.3 Auswahl an Definitionen bzw. Beschreibungen des Begriffes ERP-System	9
2.4 Merkmale eines ERP-Systems	10
2.5 Unterscheidungsmerkmale Geschäftsprozess, Workflow	16
2.6 Workflow-Arten nach Strukturierungsgrad	17
2.7 Stufen der Anwendungsintegration	32

IV. Abkürzungsverzeichnis

(e)EPK	(erweiterte) Ereignisgesteuerte Prozesskette, Seite 15
API	Application Programming Interface, Seite 41
BAM	Business Activity Monitoring, Seite 42
BMWi	Bundesministerium für Wirtschaft und Energie, Seite 5
BPMN	Business Process Model and Notation, Seite 15
CDI	Contexts and Dependency Injection, Seite 41
EJB	Enterprise Java Bean, Seite 37
ERP	Enterprise Resource Planning, Seite 1
EU	Europäische Union, Seite 5
IDE	Integrated Development Environment, Seite 40
JDBC	Java Database Connectivity, Seite 52
JMS	Java Message Service, Seite 41
KMU	Kleine und mittlere Unternehmen, Seite 1
O/R	Objekt/Relational, Seite 52
OCR	Optical Character Recognition, Seite 11
ODBC	Open Database Connectivity, Seite 52
PDF	Portable Document Format, Seite 65
RDBMS	Relationales Datenbank Management System, Seite 35
ReST	Representational State Transfer, Seite 41
SCM	Supply Chain Management, Seite 10
SQL	Structured Query-Language, Seite 52
UML	Unified Modeling Language, Seite 15
WAPI	Workflow Application Programming Interface, Seite 22
WfMC	Workflow Management Coalition, Seite 19
WfMS	Workflow-Management-System, Seite 2
WKD	Wertschöpfungskettendiagramm, Seite 15
WS	Webservice, Seite 41

1 Einleitung

Dieses Kapitel soll einleitend die Motivation zur Bearbeitung des vorliegenden Themas und die gesteckten Ziele in entsprechendem Erfordernis vorstellen. Gleichzeitig soll eine fachlich-thematische Abgrenzung stattfinden, welche den zu erwartenden Inhalt konkretisiert. Um den genauen Aufbau der Arbeit kurz zu umreisen, folgt abschließend eine verdichtete Inhaltsangabe aller enthaltenen Kapitel.

1.1 Motivation

Der Einsatz von Enterprise-Resource-Planning-Systemen (ERP-System) ist in Deutschland weit verbreitet. Bezogen auf die Verwendung eines ERP-Systems in Abhängigkeit der Unternehmensgröße sind prozentual betrachtet lediglich marginale Unterschiede auszumachen. Die Verbreitung im Bereich der Großunternehmen sowie Kleiner und Mittlerer Unternehmen (KMU) ist nahezu übereinstimmend, wie Tabelle 1.1 zeigt.

Tabelle 1.1: Verbreitung von ERP-Systemen in Deutschland (stand 2011)¹

Art des ERP-Systems	50-99 MA*	100-499 MA*,**	>500 MA*	Gesamt
ohne ERP-System	12,2 %	7,4 %	7,6 %	8,4 %
Standardsystem	73,3 %	78,3 %	80,0 %	78,2 %
Eigenentwicklung	31,7 %	32,1 %	37,0 %	33,1 %
Spezial- bzw. Einzellösung	17,9 %	18,5 %	24,5 %	21,0 %

Legende:

* Die Abkürzung MA steht für Mitarbeiter.

** Als KMU bezeichnet man Unternehmen mit bis zu 249 beschäftigten Mitarbeitern (siehe auch Abschnitt 2.1), weshalb die Angaben nur anteilig den KMU zuzuordnen sind.

Sehr deutlich zu erkennen ist der überwiegende Einsatz von ERP-Standardsystemen. Immanent für Standardsysteme ist, dass sie die Wertschöpfungsprozesse eines Anwenderunternehmens nur bedingt abbilden. Sie sind als Generalisten anzusehen,² welche allgemeine Wertschöpfungsprozesse einer bestimmten Branche integrieren. Durch den Einsatz von ERP-Standardsystemen erhalten Anwenderunternehmen demnach nur einen geringen Wettbewerbsvorteil. Dies führt gemeinhin zu Anpassungen des ERP-Systems beziehungsweise zur parallelen Verwendung von Legacy-Systemen (dt. Altsysteme) oder zur Anschaffung zusätzlicher Softwaresysteme, welche die Bedürfnisse

¹ vgl. [Gmb11, S. 17 f.] Die Tabelle bildet eine Zusammenfassung einzelner Diagramme. Eine Übertretung der 100%-Marke ist auf eine parallele Nutzung verschiedener ERP-Systemarten (z. B. Standardsystem und Speziallösung) zurückzuführen.

² vgl. [SM13, S. 244]

des Anwenderunternehmens in speziellen Bereichen besser abdecken. Daraus ergeben sich sogleich neue Probleme. Hierzu zählen etwa Medienbrüche, die zu Ablaufunterbrechungen oder -verzögerungen innerhalb des Wertschöpfungsprozesses führen, deren Ursache u. a. auf verlorene bzw. fehlerhafte Daten und einer notwendigen Doppel- oder Mehrfacherfassung zurückzuführen sind.³

Darüber hinaus stellt die stetig steigende Komplexität der Wertschöpfungsprozesse ein weiteres Problem dar.⁴ Dies erfordert abermals eine ständige Anpassung des ERP-Standardsystems. Für Entwicklerunternehmen bedeutet dies einen erhöhten Bedarf an Mitarbeiterressourcen aufgrund gewünschter Anpassungen des Anwenderunternehmens. Stehen in diesem Zusammenhang keine freien Mitarbeiterressourcen zur Verfügung, ist eine Weiterentwicklung des ERP-Systems – besonders für Entwicklerunternehmen mit geringer Mitarbeiterzahl – nur bedingt möglich oder gar unmöglich. Ein weiterer wichtiger Punkt ist, dass die stetig steigende Komplexität der Wertschöpfungsprozesse eine effiziente Kontrolle eben dieser erschwert.⁵ Dies hat zur Folge, dass kritische Bereiche innerhalb des Wertschöpfungsprozesses nur schwer zu identifizieren sind und demnach notwendige Optimierungen ausbleiben.

1.2 Ziel

Workflow-Management-Systeme (WfMS) können dazu beitragen den oben beschriebenen Problemen entgegenzuwirken. Dabei betreffen dessen Vorzügen nicht ausschließlich Wertschöpfungsprozesse. Vielmehr kann die Gesamtheit aller Unternehmensprozesse – im Folgenden allgemein als Geschäftsprozesse bezeichnet – vom Einsatz eines Workflow-Management-Systems profitieren.

Durch die Integration eines Workflow-Management-Systems in das bestehende ERP-System *hscERP* der Firma *hsc solutions*⁶ sollen eben diese Vorzüge Bestandteil des hauseigenen ERP-Systems werden. In einem ersten Schritt wird daher das Ziel verfolgt, ausgewählte Funktionalitäten des Workflow-Management-Systems in das ERP-System *hscERP* zu übertragen. Ziel der Integration ist aus Anwendersicht die Nutzung von Funktionen des Workflow-Management-Systems, ohne auf ein alternatives Softwaresystem zurückgreifen zu müssen. Aus Entwicklersicht soll die Integration zukünftig dazu beitragen den Anpassungsaufwand des ERP-Systems zu verringern, was sogleich eine verstärkte Konzentration auf die Weiterentwicklung des ERP-Systems begünstigt. Ein Austausch des Workflow-Management-Systems soll in diesem Zusammenhang zu jeder Zeit möglich sein.

³ vgl. [SM13, S. 19]

⁴ vgl. [SM13, S. 16]

⁵ vgl. [SM13, S. 18]

⁶ Nähere Informationen zur Firma können über www.hsc-solutions.de bezogen werden.

1.3 Abgrenzung

Die vorliegende Arbeit beschäftigt sich daher mit dem Thema Software- bzw. Systemintegration im Bereich kleiner und mittlerer Unternehmen sowie in Auszügen mit dem Themenbereich des Geschäftsprozess-Managements. Den Kern der Arbeit bildet hierbei vorrangig die Systemintegration. Das Geschäftsprozess-Management sowie damit verbundene Themen, wie beispielsweise die Analyse, Modellierung oder Optimierung von Geschäftsprozessen finden in dieser Arbeit lediglich beiläufig Erwähnung. Eine tiefgreifende Auseinandersetzung mit dem Thema findet demnach nicht statt. Die in dieser Arbeit abgebildeten Geschäftsprozessmodelle dienen daher einzig zum besseren Verständnis des aktuell dargelegten Sachverhalts und stellen für die Integration eine untergeordnete Rolle dar.

Bei der Integration des Workflow-Management-Systems gilt es ein Stadium zu erreichen, welches den produktiven Einsatz bei einem Anwenderunternehmen erlaubt. Dies impliziert, dass die Funktionsintegration des Workflow-Management-Systems in das bestehende ERP-System *hscERP* abgeschlossen ist und eine weitestgehend problemfreie Verwendung der integrierten Funktionen garantiert werden kann. Gleichwohl ist es nicht das Ziel der Arbeit, alle Funktionen des Workflow-Management-Systems in das ERP-System zu übertragen. Die Arbeit wird sich daher auf Management- und Analysefunktionen beschränken.

Des Weiteren ist eine Auseinandersetzung mit verschiedenen Workflow-Management-Systemen nicht Gegenstand der vorliegenden Arbeit. Das zu integrierende Workflow-Management-System *jBPM* galt von Beginn an als Favorit. Diesbezüglich wurde von einer Vergleichsanalyse verschiedener Workflow-Management-Systeme abgesehen.

1.4 Aufbau

Das aktuelle Kapitel 1 Einleitung, stellte die Motivation zur Anfertigung der vorliegenden Arbeit kurz vor. Gleichzeitig wurde eine Beschreibung der zu erreichenden Ziele sowie eine klare Abgrenzung zu angrenzenden Themen vorgenommen. Nachfolgende Absätze sollen Aufschluss über den weiteren Aufbau der Arbeit geben und den zu erwartenden Inhalt kurz umreisen.

Kapitel 2 widmet sich der Entwicklung eines einheitlichen Fachterminus. Es werden Abhängigkeiten der häufig synonym verwendeten Begriffe aufgezeigt und gleichzeitig eine Abgrenzung zwischen den verschiedenen Begriffen vorgenommen. Im Ergebnis soll dieses Kapitel zu einem konkreten Verständnis in Bezug auf die Funktionsweise der betrachteten Softwaresysteme führen. Gleichzeitig definiert es die Grenzen des späteren Integrationsvorhabens, indem es Integrationsmöglichkeiten für Workflow-Management-Systeme aufzeigt.

In Kapitel 3 erfolgt eine detaillierte Analyse des existierenden Enterprise Resource Planning Systems sowie dem zu integrierenden Workflow-Management-System. Grundlage der Analyse bildet das Wissen über die Systeme aus dem vorangegangenen Kapitel. Die aus der Analyse gewonnenen Erkenntnisse werden am Ende verwendet, um Anforderungen an das Integrationskonzept zu definieren.

Die Entwicklung eines Integrationskonzeptes erfolgt in Kapitel 4. Es enthält eine abstrakte Beschreibung aller benötigten Systemkomponenten sowie deren Kommunikation untereinander. Der Rahmen wird hierbei durch die Client-Server-Architektur des Enterprise Resource Planning Systems vorgegeben.

Kapitel 5 beschäftigt sich anschließend mit der Umsetzung des Integrationskonzeptes am Beispiel des ERP-Systems *hscERP* und dem Workflow-Management-System *jBPM*. In Form einer prototypischen Implementierung werden die zentralen Funktionen, welche für den Betrieb eines Workflow-Management-Systems benötigt werden, implementiert. Die Funktionsfähigkeit wird am Ende anhand eines ausgewählten Beispiels verifiziert.

Den Abschluss der Arbeit bildet Kapitel 6, in welchem eine kritische Auseinandersetzung mit den gesteckten und tatsächlich erreichten Zielen vorgenommen wird. Einen zusätzlichen Punkt bildet der Ausblick, welcher Vorschläge und Anregungen für den weiteren Ausbau des Workflow-Management-Systems beinhaltet.

2 Grundlagen

Im Folgenden werden die in der Arbeit verwendeten Begrifflichkeiten genauer definiert. Gleichzeitig gilt es relevante Merkmale zu identifizieren, welche im späteren Verlauf bei der Konzeption zu berücksichtigen sind. Um den allgemeinen Anspruch der Arbeit zu genügen, geschieht dies ungeachtet des unternehmensspezifischen Hintergrundes innerhalb der Firma *hscERP*.

2.1 Kleine und mittlere Unternehmen

Nach Angaben des Instituts für Mittelstandsforschung Bonn (IfM Bonn) bilden Kleine und mittlere Unternehmen (KMU) mit über 99 % (3,65 Mio. Unternehmen) fast die Gesamtheit aller in Deutschland ansässigen Wirtschaftsunternehmen.⁷ Einen genauen Überblick der Unternehmensverteilung, bezogen auf die Anzahl sozialversicherungspflichtig Beschäftigter, verschafft Abbildung 2.1.

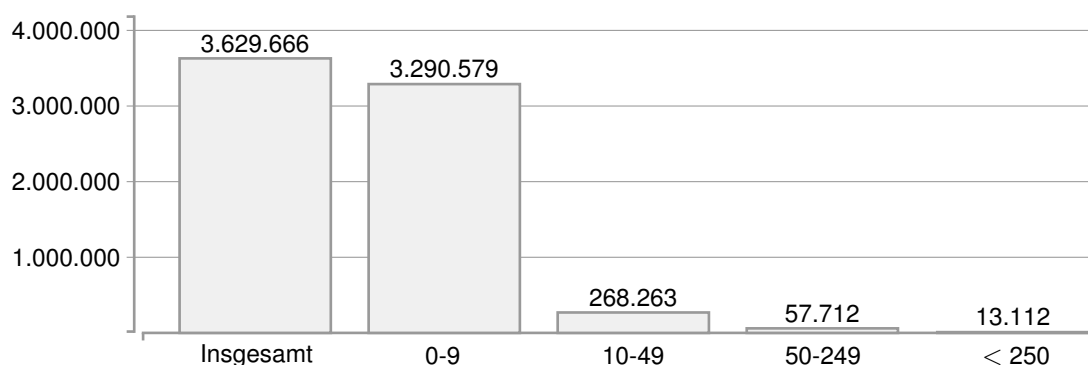


Abbildung 2.1: Anzahl der KMU in Deutschland nach Beschäftigtenzahl (Stand 2015)⁸

Unternehmen mit über 249 Beschäftigten zählen nicht zu den KMU, wie nachfolgende Begriffsdefinition zeigt und sind daher in der Gesamtstatistik nicht zu berücksichtigen.

2.1.1 Begriffsdefinition

Nach einer Definition der Europäischen Union⁹ (EU) – unter Bezugnahme auf das Bundeswirtschaftsministerium für Wirtschaft und Energie (BMWi) – zählen zu den KMU

⁷ vgl. [fMB16, Abschnitt: Unternehmensverteilung]

⁸ vgl. [Gmb15] Die Statistik bezieht sich auf Unternehmen mit steuerbarem Umsatz aus Lieferungen und Leistungen und/ oder mit sozialversicherungspflichtig Beschäftigten im Berichtsjahr 2013.

⁹ vgl. [Kom03]

Kleinst- und Kleinunternehmen sowie mittlere Unternehmen. Die Mitarbeiterhöchstgrenze für KMU beträgt 249 Beschäftigte. Zusätzlich gilt es die Vorgabe eines maximalen Jahresumsatzes von 50 Mio. EUR oder einer maximalen Bilanzsumme von 43 Mio EUR. nicht zu übersteigen. Alle Unternehmen, die diese Werte nicht bzw. nur mit geringen Abweichungen überschreiten, zählen in Deutschland¹⁰ zu den KMU. Tabelle 2.1 veranschaulicht diesen Sachverhalt noch einmal und unterscheidet dabei genau zwischen Kleinst- und Kleinunternehmen sowie mittleren Unternehmen.

Tabelle 2.1: Klassifizierung Klein- und Mittelunternehmen der EU¹¹

Unternehmenskategorie	Mitarbeiter	Umsatz (EUR)	Bilanzsumme (EUR)
Mittleres Unternehmen	50-249	≤ 50 Mio.	43 Mio.
Kleinunternehmen	10-49	≤ 10 Mio.	10 Mio.
Kleinstunternehmen	1-9	≤ 2 Mio.	2 Mio.

Bereits anhand der beschriebenen quantitativen Merkmale lässt sich der Begriff KMU festmachen und gestattet eine Abgrenzung gegenüber Großunternehmen. Daneben existieren weitere Merkmale, welche helfen den Begriff KMU sowie dessen Besonderheiten zu präzisieren.

2.1.2 Qualitative Merkmale

Die oben genannten quantitativen Merkmale beschränken sich vorwiegend auf Zahlenwerte, wobei eine Abgrenzung besonders an der Schwelle zu Großunternehmen schwierig vorzunehmen ist. Qualitative Merkmale versuchen diesen Missstand auszugleichen, indem sie für KMU charakteristische Merkmale zur Abgrenzung heranziehen. Einen Ausschnitt aus dem Merkmalskatalog von Pfohl liefert Tabelle 2.2.

Tabelle 2.2: Qualitative Merkmalsunterscheidung zwischen KMU und Großunternehmen¹²

Klein- und Mittelbetriebe (KMU)		Großunternehmen
Unternehmensführung		
U1	Eigentümer-Unternehmen	Manager
U2	Patriarchalische Führung	Führung nach Management-by-Prinzipien
U3	Große Bedeutung von Improvisation und Intuition	Geringe Bedeutung von Improvisation und Intuition
U4	Kaum Gruppenentscheidungen	Häufig Gruppenentscheidungen

¹⁰ Die Kriterien für die Zuordnung sind eine Empfehlung der Kommission vom 6. Mai 2003.

¹¹ vgl. [IS14, S. 19]

Tabelle 2.2: Qualitative Merkmalsunterscheidung zwischen KMU und Großunternehmen (Fortsetzung)

	Klein- und Mittelbetriebe (KMU)	Großunternehmen
U5	Kaum Planung	Umfangreiche Planung
U6	Durch Funktionshäufung überlastet, wenn Arbeitsteilung, dann personenbezogen	Hochgradig, sachbezogene Arbeitsteilung
U7	Führungspotential nicht austauschbar	Führungspotential austauschbar
U8	Unzureichendes Informationswesen zur Nutzung vorhandener Flexibilitätsvorteile	Ausgebautes, formalisiertes Informationswesen
Organisation		
O1	Funktionshäufung	Arbeitsteilung
O2	Kaum Abteilungsbildung	Umfangreiche Abteilungsbildung
O3	Kurze direkte Informationswege	Vorgeschriebene Informationswege
O4	Starke persönliche Bindung	Geringe persönliche Bindung
O5	Weisung und Kontrolle im direkten personenbezogenen Kontakt	Formalisierte unpersönliche Weisungs- und Kontrollbeziehungen
O6	Delegation im beschränkten Umfang	Delegation in vielen Bereichen
O7	Kaum Koordinationsprobleme	Große Koordinationsprobleme
O8	Geringer Formalisierungsgrad	Hoher Formalisierungsgrad
O9	Hohe Flexibilität	Geringe Flexibilität
Personal		
P1	Geringe Anzahl von Beschäftigten	Hohe Anzahl von Beschäftigten
P2	Häufig unbedeutender Anteil von ungelernten und angelernten Arbeitskräften	Häufig großer Anteil von ungelernten und angelernten Arbeitskräften
P3	Wenig Akademiker beschäftigt	Akademiker in größerem Umfang beschäftigt
P4	Überwiegend breites Fachwissen	Starke Tendenz zum Spezialisten

¹² vgl. [PPA⁺13, S. 19 ff.], Die Merkmale in der Tabelle stellen lediglich einen Auszug der für diese Arbeit wesentlichen Merkmale dar. Zur besseren Unterscheidung der Merkmale wurde sich dafür entschieden zusätzlich eine Nummerierung einzuführen.

Pfohl liefert den umfangreichsten in der Literatur zu findenden Merkmalskatalog für KMUs. Daneben existieren weitere Merkmalskataloge, die sich in ihrem Ausmaß stark unterscheiden und untereinander nicht widerspruchsfrei sind.¹³ Dabei ist anzumerken, dass bereits die in Tabelle 2.2 gelisteten Merkmale nicht alle innerhalb eines KMUs zwingend in der Praxis anzutreffen sind.¹⁴ Die Summe aller Merkmale dient primär zur Unterstützung der Abgrenzung, wobei eine hohe Quote an Übereinstimmungen mit hoher Wahrscheinlichkeit auf ein KMU hindeutet. Der in Tabelle 2.2 zu sehende Auszug deckt die für diese Arbeit relevanten Merkmale ab und soll an dieser Stelle genügen. Im anschließenden Kapitel sind daraus allgemeine Anforderungen abzuleiten, welche es bei der Konzeption zu berücksichtigen gilt.

2.2 ERP-Systeme für KMU

Die ersten ERP-Systeme entstanden in den 80er-Jahren mit dem Fokus, Geschäftsdaten und Funktionen aller Unternehmensbereiche zu integrieren.¹⁵ Dabei galt es verschiedenen Softwaresysteme eines Anwenderunternehmens über eine gemeinsame Datenbasis – in Form einer einheitlichen Datenbank – miteinander zu koppeln und somit einen Daten- und Informationsaustausch zwischen den Systemen zu gewährleisten.¹⁶ Mit der Zeit rückte eine prozessorientierte Sichtweise in den Vordergrund. Infolgedessen galt es verstärkt die Geschäftsprozesse eines Anwenderunternehmens mit Hilfe des ERP-Systems abzubilden.¹⁷ Aus den vorrangig als Individuallösung für Anwenderunternehmen konzipierten Systemen entstanden zunehmend ERP-Standardsysteme, welche auf die innerbetrieblichen Geschäftsprozesse eines Anwenderunternehmens anpassbar waren.¹⁸

2.2.1 Begriffsdefinition

Trotz der weiten Verbreitung von ERP-Systemen findet man in der Literatur keine einheitliche Begriffsdefinition. Dies kann unter anderem mit einem sich über die Zeit veränderten Verständnis hinsichtlich der Aufgaben eines ERP-Systemes begründet werden. Dennoch lassen sich in den verschiedenen Definitionen bzw. Umschreibungen der Autoren mehrere Gemeinsamkeiten erkennen. In Tabelle 2.3 (auf Seite 9) befinden sich drei selbst ausgewählte Definitionen bzw. Beschreibungen des Begriffs ERP-System. Auf deren Grundlage ist im Folgenden eine geeignete Begriffsdefinition zu finden.

¹³ vgl. [IS14, S. 20]

¹⁴ ibidem vgl. [IS14, S. 20]

¹⁵ vgl. [TW15, S. 71]

¹⁶ vgl. [Mer12, S. 13 f.]

¹⁷ vgl. [TW15, S. 71]

¹⁸ ibidem vgl. [TW15, S. 71]

Tabelle 2.3: Auswahl an Definitionen bzw. Beschreibungen des Begriffes ERP-System

Autor	Definition bzw. Beschreibung
Nach Leimeister	Nach Leimeister handelt es sich um ein ERP-System, wenn „ein integriertes Gesamtsystem alle wesentlichen operativen Funktionen und Führungsfunktionen unterstützt“. ¹⁹
Nach Gadatsch	Für Gadatsch ist unter einem ERP-System „ein Softwaresystem zu verstehen, bei dem mehrere betriebswirtschaftliche Standard Business-Applikationen durch eine gemeinsame Datenbasis integriert sind“. ²⁰ Zu den Standard Business-Applikationen gehören nach Gadatsch u. a. Anwendungen aus den Bereichen: ²¹ <ul style="list-style-type: none"> • Beschaffungsmanagement • Lagerhaltung • Produktion • Distribution/Transport • Human Resources und • Lohnbuchhaltung.
Nach Meyer	Meyer bezeichnet ein ERP-System als ein hochintegriertes betriebswirtschaftliches Informationssystem zur Abbildung aller Geschäftsprozesse der Hauptfunktionsbereiche eines Unternehmens sowie der Unterstützung aller begleitenden Prozesse (z. B. Anlagenbuchhaltung) und Schnittstellen zu Kunden und Lieferanten. ²²

Allen Definitionen ist gemein, dass sie den Faktor Integration enthalten sowie eine betriebswirtschaftliche Ausrichtung des ERP-Systems implizieren. Die alleinige Reduzierung auf diese beiden Kriterien wird heutigen ERP-Systemen jedoch kaum gerecht. Daher wird im Folgenden eine Definition gewählt, welche die einzelnen Kriterien teilweise vereint.

Ein ERP-System ist demnach ein Softwaresystem zur Unterstützung der innerbetrieblichen operativen und dispositiven Geschäftsprozesse eines Unternehmens sowie verschiedener Schnittstellen zu Kunden und Lieferanten. Hierbei muss klar zwischen einem einfachen Datenaustausch und einer gemeinsamen Koordinierung der einzelnen Anwendungssystem hinsichtlich der eingesetzten Methodik unterschieden werden. ²³ Letzteres ist beispielsweise durch die Anbindung eines Supply-Chain-Management-

¹⁹ [Lei15a]

²⁰ [Gad12, S. 261]

²¹ ibidem vgl. [Gad12, S. 306]

²² vgl. [Mey15, Folie 25]

²³ vgl. [Lei15a, S. 175]

Systems (SCM-System) zu erreichen. Aufgrund der fehlenden Relevanz für die nachfolgenden Betrachtungen soll an dieser Stelle jedoch nicht näher auf dessen Bedeutung eingegangen werden.

2.2.2 Merkmale beziehungsweise Bestandteile heutiger ERP-Systeme

Alle oben genannten Definitionen erlauben einen großen Deutungsspielraum. So bleibt beispielsweise vollkommen offen, was im Speziellen unter Integration zu verstehen ist. Im Folgenden sollen daher konkrete Merkmale eines ERP-Systems genannt werden, welche eine bessere Einordnung des Begriffes ermöglichen.

Eine übersichtliche Liste von Merkmalen bietet Gadatsch. Dieser fasst sechs wesentliche Merkmale von ERP-Systemen zusammen, welche in Tabelle 2.4 auf dieser und der Folgeseite festgehalten sind.

Tabelle 2.4: Merkmale eines ERP-Systems²⁴

Merkmale	Kurzbeschreibung	Beispiel
1. Datenintegration	Mehrere Softwaremodule nutzen gemeinsame Daten	Ein Vertriebs- und Buchhaltungsmodul verwendet jeweils die Kundendaten.
2. Prozessintegration	Abteilungsübergreifende Geschäftsprozesse werden durch mehrere beteiligte Softwaremodule gemeinsam unterstützt.	Die Kundenauftragsbearbeitung wird vom Eingang der Kundenanfrage, über die Fertigung bis hin zur Auslieferung und Bezahlung der Waren mit Hilfe mehrerer Softwaremodule (Vertriebsabwicklung, Produktionsplanung, Versand, Finanzen) unterstützt.
3. Operative Funktionalitäten	Unterstützung operativer Aufgaben eines Unternehmens zur Abwicklung von Geschäftsvorfällen.	Auftragsbearbeitung, Fertigungsplanung, Kundenbuchhaltung, Erfassen Eingangsrechnung, Gehaltsabrechnung.
4. Einheitliches Entwicklungskonzept	Softwaremodule nutzen gemeinsames Repository und basieren auf einheitlichen Entwicklungsstandards	Gleiches Bildschirmmaskenlayout, gleichartige Fehlermeldungen.

Tabelle 2.4: Merkmale eines ERP-Systems (Fortsetzung)

Merkmale	Kurzbeschreibung	Beispiel
5. Schichtenarchitektur	Softwarearchitektur zur Unterstützung einer über mehrere Abteilungen und Standorte, ggf. auch Länder, verteilten Verarbeitung.	Client-/Server-Architektur zur Realisierung des dezentralen Zugriffs auf Daten und Funktionen
6. Transaktionsorientierung	Onlineverarbeitung von Geschäftsvorfällen und Speicherung der Daten auf Datenbank	Anlegen eines Kundenauftrags, Buchen einer Eingangsrechnung.

Zu den Bestandteilen eines ERP-Systemes zählt in der Regel ein Basissystem sowie funktionsbezogene Module.²⁵ So beherbergen ERP-Systeme beispielsweise Module für die Beschaffung, Materialwirtschaft, das Projektmanagement, den Einkauf und Vertrieb, sowie der Produktionsplanung und -steuerung eines Unternehmens. Das Basissystem enthält modulübergreifende Funktionalitäten, wie die Funktion zum Drucken einer PDF, das Anmelden am System oder den Datenbankzugriff. Die einzelnen Module basieren hierbei auf einer einheitlichen Datenbank und sind funktional so weit integriert, dass bereichsübergreifende Geschäftsprozesse abbildbar sind.²⁶ Dies bedeutet, dass verteilt auf die einzelnen Module alle Funktionen enthalten sind, um den aktuellen Geschäftsprozess erfolgreich zu bearbeiten. Die Funktionsintegration reduziert hierbei die Anzahl vorhandener Schnittstellen und fördert somit die einmalige Erfassung von Geschäftsdaten.²⁷ Die Unterstützung operativer Funktionalitäten bspw. der automatischen Gehaltsabrechnung bzw. Rechnungserfassung (mittels Optical Character Recognition (OCR)) gestattet eine zeitliche Optimierung der Geschäftsprozesse und bietet die Möglichkeit freiwerdende Mitarbeiterressourcen umzuverteilen. Ein weiteres Merkmal bildet die Etablierung eines einheitlichen Entwicklungskonzeptes. Dies zeichnet sich vor allem durch ein homogenes Benutzerlayout, Bedienkonzept und einem durchgängig identischen Verhalten im Fehlerfall aus. Hierdurch wird die Akzeptanz durch die Mitarbeiter innerhalb des Anwenderunternehmens erhöht und eine schnelle Einarbeitung in neue Funktionalitäten des ERP-Systems gefördert. Aus den beschriebenen Merkmalen ergeben sich verschiedene Vorteile für eine Anwenderunternehmen durch den Einsatz eines ERP-System, welche zum Teil bereits angesprochen wurden. Hinzu kommen Nachteile, welche auf den ersten Blick nicht immer sofort ersichtlich sind. Für eine spätere Gegenüberstellung von ERP- und Workflow-Management-System werden im Folgenden Unterabschnitt die Vor- und Nachteile aufgezeigt.

²⁴ [Gad12, S. 262]

²⁵ vgl. [Lei15a, S. 172]

²⁶ ibidem vgl. [Lei15a, S. 172]

²⁷ vgl. [Gad12, S. 263]

2.2.3 Vor- und Nachteile durch den Einsatz eines ERP-Systems

Durch den Einsatz eines ERP-Systemes und der damit verbundenen rechnergestützten Abarbeitung einzelner Arbeitsschritte innerhalb des Geschäftsprozesse lassen sich einige Vorteile erzielen. Durch Eingabemasken, welche den Benutzer bspw. auf fehlende Informationen bei der Eingabe von Geschäftsdaten hinweisen, wird die Fehleranfälligkeit reduziert und die Datenkonsistenz erhöht. Gleichzeitig sind damit Kosten- und Zeitersparnisse im Betriebsablauf verbunden,²⁸ welche u. a. auf konsistente Daten in allen Arbeitsschritten und einer optimierten Lagerverwaltung zurückzuführen sind. Aufgrund der gemeinsamen Datenbank sowie der Unterstützung operativer und dispositiver Funktionen vom Einkauf bis zum Verkauf der Produkte bzw. Dienstleistungen bieten ERP-Systeme eine stetige Verfügbarkeit aktueller Informationen bezüglich Umsatz, Kosten, Durchlaufzeiten, etc.²⁹ In Anbetracht einer optimierten Lagerverwaltung gestatten ERP-Systeme Materialengpässe schnell aufzudecken und gewähren eine optimale bedarfsgerechte Produktion von Waren. Eine derartig optimierte Produktion kann zu einer Erhöhung der Kundenzufriedenheit beitragen, indem eine umgehende Warenlieferung forciert werden kann.

Neben den Vorteilen existieren verschiedene Nachteile, welche häufig mit dem Einsatz eines ERP-Systems einhergehen. Hierzu gehören hohe Investitionskosten bei der Erstanschaffung.³⁰ Gerade im KMU-Bereich fällt die Risikobereitschaft für Investitionen, aufgrund der persönlichen Haftung gegenüber Großunternehmen, jedoch deutlich geringer aus.³¹ Gleichzeitig ist die Einführung eines ERP-Systems stets mit einer Neugestaltung vorhandener Geschäftsprozesse verbunden. Die Ursache liegt primär darin begründet, dass Entwicklerunternehmen allgemeine Annahmen über den optimalen Ablauf der Geschäftsprozesse innerhalb der entsprechenden Branche aufstellt, die sogleich als Vorgabe über das ERP-System abgebildet werden.³² Kritisch ist in diesem Zusammenhang vor allem eine notwendige Anpassung der Kernprozesse. Diese führen im Allgemeinen zu einem Wettbewerbsvorteil des Anwenderunternehmens und sind daher erfolgsentscheidend. Durch Anpassungen des ERP-Systems auf die Kernprozesse des Anwenderunternehmens ist dieser Nachteil zwar auszugleichen, die Kosten hierfür sind allerdings abermals vom Anwenderunternehmen zu tragen.³³

²⁸ vgl. [Lei12, S. 50 ff.]

²⁹ ibidem vgl. [Lei12, S. 50]

³⁰ vgl. [Lei15a, S. 72]

³¹ vgl. [Lei12, S. 309]

³² ibidem vgl. [Lei12, S. 61 f.]

³³ vgl. [Lei15a, S. 72 f.]

2.3 Workflow-Management-Systeme für KMU

Die Aufgabenbereiche von Workflow-Management-Systemen haben eine starke Ähnlichkeit mit denen von ERP-Systemen. Beide unterstützen die Neugestaltung, Integration und das Management von Geschäftsprozessen. Konkrete Unterschiede sind oftmals erst bei genauerer Betrachtung festzustellen. Dieser Abschnitt widmet sich daher der detaillierten Auseinandersetzung mit dem Thema Workflow-Management-System.

2.3.1 Grundlegende Begrifflichkeiten

Bevor eine Auseinandersetzung mit dem Begriff Workflow-Management-System erfolgen kann, sind jedoch zunächst einzelne Begrifflichkeiten näher zu erläutern. Dies soll zu einem besseren Verständnis der im Anschluss folgenden Ausführungen zum Thema Workflow-Management-System beitragen. Zu den grundlegenden Begrifflichkeiten zählen:

- Geschäftsprozess
- Geschäftsprozessmodell
- Workflow
- Workflowmodell sowie
- Workflow Management.

Geschäftsprozess

Im betriebswirtschaftlichen Umfeld existiert eine Vielzahl verschiedener Geschäftsprozesse. Hierzu gehören u. a. *Kreditantrag stellen*, *Kundenauftrag bearbeiten* (vom Angebot bis zur Auslieferung) oder *Kundenreklamation prüfen*. Eine ausführliche Definition des Begriffes liefert Gadatsch. Dieser definiert einen Geschäftsprozess als

„eine zielgerichtete, zeitlich-logische Abfolge von Aufgaben, die arbeitsteilig von mehreren Organisationen oder Organisationseinheiten unter Nutzung von Informations- und Kommunikationstechnologien ausgeführt werden können. Er dient der Erstellung von Leistungen entsprechend der vorgegebenen, aus der Unternehmensstrategie abgeleiteten Prozesszielen“.³⁴

Ausgehend von einem Startereignis, begleitet durch ein beendendes Endereignis, erzeugt ein Geschäftsprozess demnach stets einen auf Basis der Unternehmensstrategie im Vorfeld definierten Output (z. B. Produkt, Dienstleistung). Für dessen Erzeugung sind im Allgemeinen eine Menge unterschiedlicher Inputs (z. B. Anschrift des Kunden,

³⁴ [Gad12, S. 36]

Bestellnummer oder Kundennummer) notwendig. Einzelne Aufgaben innerhalb des Geschäftsprozesses werden häufig auch als Aktivität oder Task bezeichnet, welche in logischer Verbindung stehen.³⁵ Die Ausführung der Aktivitäten erfolgt durch über Rollen zugewiesene Akteure, jene zusätzliche Inputs liefern und hierarchie- und standortübergreifend verteilt sein können.³⁶ Bei den Akteuren unterscheiden man zwischen Mitarbeitern und Fremdsystemen. Exemplarisch für ein Fremdsystem ist beispielsweise ein an das ERP-System angeschlossenes Dokumenten-Management-System mit OCR-Funktionalität, welches zum Beispiel zur automatischen Ermittlung von Bankdaten für das Lastschriftverfahren dienen kann.

Die formale Beschreibung „kann auf unterschiedlichen Detaillierungsebenen und aus mehreren Sichten“³⁷ erfolgen. Der höchste Detaillierungsgrad ist erreicht, wenn eine einzelne Aktivität von einem einzelnen Mitarbeiter in einem einzelnen Arbeitsschritt ohne einen Arbeitsplatzwechsel ausführbar ist.³⁸ Zur visuell-strukturellen Beschreibung eines Geschäftsprozesses existieren abhängig von der notwendigen Sicht und dem Detaillierungsgrad verschiedene Geschäftsprozessmodelle.

Geschäftsprozessmodell

Geschäftsprozessmodelle beschreiben den Geschäftsprozess allgemein in einer für den Menschen leicht verständlichen, visuell-strukturellen Form und dienen der Dokumentation und unterstützen die Analyse sowie Gestaltung von Geschäftsprozessen. Synonyme Bezeichnungen sind u. a. Prozessdefinition, Prozess-Schema, Prozessmodell.³⁹ Nach Hagen beschreibt ein Geschäftsprozessmodell

„die Struktur eines realen (...) [Geschäftsprozesses]. Es bestimmt alle möglichen Pfade entlang des (...) [Geschäftsprozesses] und bestimmt die Regeln für die Wahl der Pfade. Weiterhin bestimmt das (...) [Geschäftsprozessmodell] alle Aktivitäten, die ausgeführt werden müssen.“⁴⁰

Die visuelle Darstellungsform erlaubt zugleich einen leichteren Zugang und kann das Verständnis der am Geschäftsprozess beteiligten Personen fördern. Weiterhin ermöglicht es eine einfache Identifikation vorhandener Mängel im Ablauf (z. B. redundante, fehlende bzw. nie erreichbare Aktivitäten). Eine derartige Identifikation ist bei einer alternativen Beschreibung eines Geschäftsprozesses bspw. in Textform nur mit erhöhtem Aufwand erreichbar.

³⁵ vgl. [Mü06, S. 7]

³⁶ ibidem vgl. [Mü06, S. 7]

³⁷ [Gad12, S. 36 f.]

³⁸ ibidem vgl. [Gad12, S. 37]

³⁹ vgl. [Hag04, S. 30]

⁴⁰ ibidem [Hag04, S. 30], Der allgemeine und synonyme Begriff Prozessmodell wurde durch den Begriff Geschäftsprozessmodell ersetzt.

Für die Modellierung stehen verschiedene grafische Modellierungssprachen zur Auswahl, welche sich in ihrer Modellierungstiefe und Ausdrucksstärke stark unterscheiden. Zu nennen sind hier beispielsweise:

- das Wertschöpfungskettendiagramm (WKD),
- die (erweiterte) Ereignisgesteuerte Prozesskette ((e)EPK),
- die Business Process Model and Notation (BPMN) sowie
- das UML Aktivitätsdiagramm.

Von den oben aufgezählten Modellierungssprachen sind lediglich die BPMN und das UML Aktivitätsdiagramm standardisiert.⁴¹ Für eine organisationsübergreifende einheitliche Nutzung eignen sich daher vornehmlich die beiden letztgenannten Modellierungssprachen. Für die nachfolgenden Betrachtungen ist ausschließlich die BPMN von Interesse. Eine ausführliche Beschreibung der übrigen – oben genannten Modellierungssprachen – bietet Gadatsch.⁴² Die Reduzierung auf BPMN wird mit dem zu integrierenden Workflow-Management-System begründet, welches bei der Modellierung auf die BPMN zurückgreift.

Workflow

Obwohl Müller darauf hinweist, dass eine „eindeutige Trennung in der ganzheitlichen Betrachtung einer Geschäftsprozessrealisierung nicht angebracht ist“⁴³, soll im Folgenden zunächst der Workflow-Begriff definiert und die Unterschiede zum Geschäftsprozessbegriff vorgestellt werden. Nach Gadatsch ist ein Workflow

„ein formal beschriebener, ganz oder teilweise automatisierter Geschäftsprozess. Er beinhaltet die zeitlichen, fachlichen und ressourcenbezogenen Spezifikationen, die für eine automatische Steuerung des Arbeitsablaufes auf der operativen Ebene erforderlich sind. Die hierbei anzustoßenden Arbeitsschritte sind zur Ausführung durch Mitarbeiter oder durch Anwendungsprogramme vorgesehen“.⁴⁴

Ein Workflow ist demnach als eine ergänzende Beschreibung zur teil- bzw. vollständigen Automatisierung eines Geschäftsprozesses anzusehen. Die fachliche Beschreibung einzelner Arbeitsschritte des Geschäftsprozesses lässt sich durch einen Workflow technisch unterstützen.⁴⁵ Ein bedeutender Unterschied liegt folglich in der Gestaltungsebene. Für eine angestrebte Automatisierung sind die zu verwendenden Systeme eindeutig zu benennen. Gleichzeitig müssen konkrete Informationen über benötigte Daten

⁴¹ beide Standards werden von der Object Management Group (OMG, www.omg.org) verwaltet.

⁴² vgl. [Gad12, S. 63 ff.]

⁴³ [Mü06, S. 14]

⁴⁴ [Gad12, S. 41]

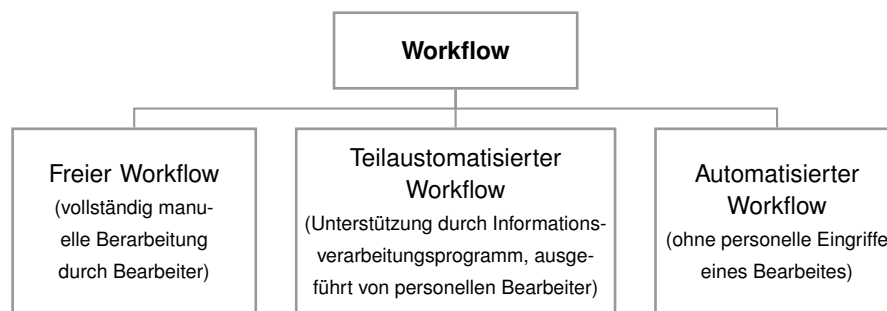
⁴⁵ [Mü06, S. 8]

und deren Verarbeitung vorliegen. Erst durch die Anreicherung um zusätzliche Informationen ist eine automatische Unterstützung umsetzbar. Der Detaillierungsgrad eines Workflows – im Gegensatz zum Geschäftsprozess – ist demnach stets höher. Nach Gadatsch sind drei grundlegende Unterscheidungsmerkmale zu konstatieren: das Ziel, die Gestaltungsebene und der Detaillierungsgrad. Tabelle 2.5 zeigt eine detaillierte Übersicht der genannten Unterscheidungsmerkmale.

Tabelle 2.5: Unterscheidungsmerkmale Geschäftsprozess, Workflow⁴⁶

	Geschäftsprozess	Workflow
Ziel	Analyse und Gestaltung von Arbeitsabläufen im Sinne gegebener (strategischer) Ziele	Spezifikation der technischen Ausführung von Arbeitsabläufen
Gestaltungsebene	Konzeptionelle Ebene mit Verbindung zur Geschäftsstrategie	Operative Ebene mit zu unterstützender Technologie
Detaillierungsgrad	In einem Zug von einem Mitarbeiter an einem Arbeitsplatz auszuführender Arbeitsschritt	Konkretisierung von Arbeitsschritten hinsichtlich Arbeitsverfahren sowie personeller und technologischer Ressourcen

Wie oben angedeutet ist der Automatisierungsgrad stark vom zugrundeliegenden Geschäftsprozess abhängig. Diesbezüglich sind drei Workflow-Arten zu unterscheiden, wie in Abbildung 2.2 zu sehen, wobei für die spätere Integration ausschließlich teilautomatisierte und automatisierte Workflows von Interesse sind.

Abbildung 2.2: Workflow-Arten nach Automatisierungsgrad⁴⁷

Eine weiteres Unterscheidungsmerkmal bildet der Strukturierungsgrad. So können Workflows wenig bis stark strukturiert sein.⁴⁸ Nach Gadatsch sind diesbezüglich drei Workflow-

⁴⁶ [Gad12, S.46]

⁴⁷ ibidem vgl. [Gad12, S. 46 f.] eigene erweiterte Darstellung.

⁴⁸ ibidem vgl. [Gad12, S. 42 ff.]

Arten zu unterscheiden, welche in Tabelle 2.6 beschrieben und anhand mindestens einem Beispiel erläutert werden.

Tabelle 2.6: Workflow-Arten nach Strukturierungsgrad⁴⁹

Workflow-Type	Beschreibung	Beispiel
Allgemeiner Workflow	Stark strukturierte Arbeitsabläufe mit Wiederholcharakter und detaillierter Spezifikation aller Ablaufschritte.	Reisekostenabrechnung, Urlaubsantragsbearbeitung, Kundenauftragsabwicklung.
Fallbezogener Workflow	Weitgehend strukturierte Arbeitsabläufe mit unstrukturierten Teilaufgaben. Individuelle Entscheidungen beeinflussen den Arbeitsablauf, einzelne Arbeitsschritte können übersprungen werden.	Kreditbearbeitung bei Banken, Schadensbearbeitung in Versicherungen, Bearbeitung von Kundenreklamationen, Einstellung von Mitarbeitern.
Ad hoc Workflow	Wenig strukturierte Aufgaben die nicht exakt im Voraus spezifizierbar sind.	Arbeitsgruppe zur Entwicklung eines Marketingkonzeptes.

Für eine Automatisierung im Sinne eines automatisierten Workflows eignen sich vor allem allgemeine Workflows. Gut strukturierte Arbeitsabläufe mit einer detaillierten Spezifikation bilden eine gute Voraussetzung für die Automatisierung. Besonders mit Blick auf die Ausführung durch ein Workflow-Management-System ist davon auszugehen, dass aufgrund des Wiederholungscharakters langfristig mit einer Minderung der Bearbeitungskosten zu rechnen ist. Demgegenüber stehen die Ad hoc Workflows, welche sich weniger bis gar nicht zur Automatisierung eignen.⁵⁰

Workflow-Modell

Geschäftsprozesse sind mittels Workflow-Management-Systemen automatisiert ausführbar. Während ein Geschäftsprozess jedoch lediglich beschreibt *Was* zu tun ist, um die Unternehmensstrategie zu unterstützen, ist das Workflow-Modell um das *Wie* für eine automatische Unterstützung zu erweitern.⁵¹ Dies kennzeichnet sogleich einen wesentlichen Unterschied zwischen Geschäftsprozessmodell und Workflow-Modell. Erreichbar ist dies durch Angabe ressourcenbezogener Spezifikationen. Hierzu zählen bspw. benötigte Informationen (z. B. Daten) innerhalb einer Aktivität bzw. für den gesamten Workflow, aber auch Informationen zu Fremdsystemen (z. B. Verbindungsinformationen) oder Kriterien für das Auslösen eines Ereignisses (engl. Event). Einzelne

⁵⁰ vgl. [Gad12, S. 43]

⁵¹ ibidem vgl. [Gad12, S. 46]

Aktivitäten können sogleich verschiedene Typen darstellen, welche manuell von einem Mitarbeiter oder automatisch beispielsweise über ein hinterlegtes Skript bzw. als Service von einem Fremdsystem zu bearbeiten sind. Abbildung 2.3 zeigt ein beispielhaftes Workflow-Modell mit einer Auswahl verschiedener Aktivitäten sowie zusätzlichen Informationen bezüglich der Datenbanknutzung einer Aktivität (engl. Task).

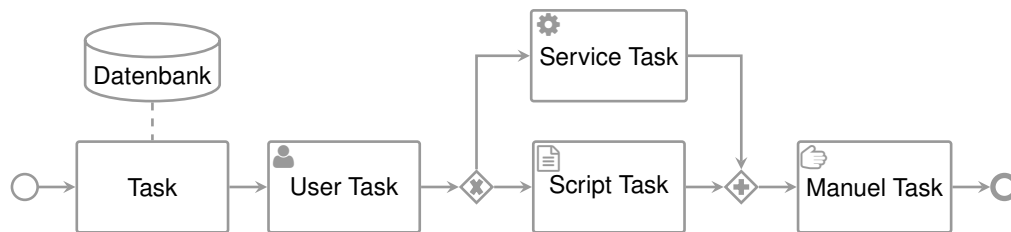


Abbildung 2.3: Beispiel eines Workflow-Modelles

Für eine spätere Ausführung der Workflow-Modelle durch ein Workflow-Management-System müssen diese in einer für die Maschine lesbaren Sprache zur Verfügung stehen. Hierfür existieren XML-basierte, standardisierte Ausführungssprachen: die XML Processing Definition Language (XPDL) und Business Process Execution Language (BPEL). BPMN in Version 2 verbindet sogleich Ausführung und Darstellung der Modelle.

Workflow Management

Das Workflow Management umfasst alle Aufgaben für die Analyse, Modellierung, Simulation, Reorganisation sowie Ausführung und Steuerung von Workflows.⁵² Es beinhaltet somit alle organisatorischen Arbeitsschritte und Abläufe für die Unterstützung eines Workflow-Regelkreis wie ihn Abbildung 2.4 darstellt.⁵³

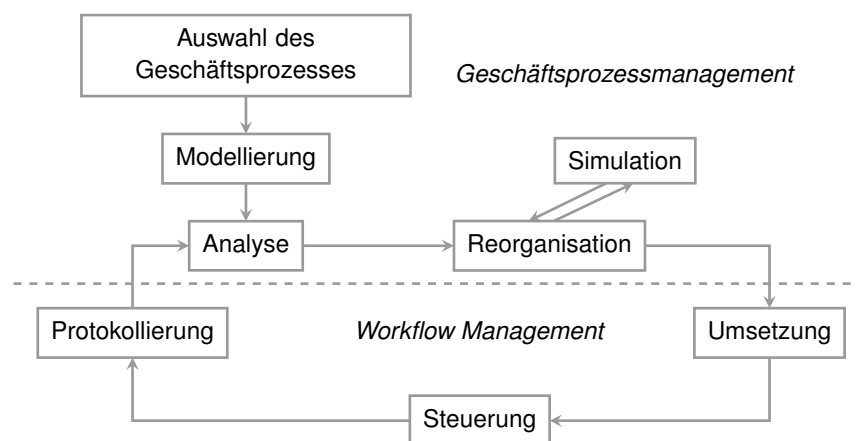


Abbildung 2.4: Workflow-Regelkreis⁵⁴

⁵² vgl. [Mü06, S. 10]

⁵³ ibidem vgl. [Mü06, S. 10]

⁵⁴ ibidem [Mü06, S. 29]

Der Workflow-Regelkreis enthält mehrere Phasen, die einmal bzw. mehrmals durchlaufen werden. Den Ausgangspunkt bildet ein konkreter Geschäftsprozess. Auf dessen Grundlage gilt es in der Phase der Modellierung das Geschäftsprozessmodell zu erstellen. In der darauf folgenden Analysephase sind u. a. die Bearbeitungs-, Transport- und Liegezeiten für einzelnen Aktivitäten innerhalb des Geschäftsprozesses zu ermitteln.⁵⁵ Gefundene Mängel im Ablauf werden in der Phase Reorganisation behoben. Das wechselseitige Zusammenspiel zwischen den Phasen Simulation und Reorganisation hilft näherungsweise die korrekten Zeiten festzustellen und die getroffenen Annahme zu überprüfen.⁵⁶ Nach einem zufriedenstellenden Simulationsergebnis folgt die Umsetzungsphase. In dieser werden die verschiedenen IT-Systeme vorbereitet und an das Workflow-Management-System angebunden. Dieses sorgt nun in der Steuerungsphase für eine automatisierte Abarbeitung der Aktivitäten, indem alle notwendigen Ressourcen bereitgestellt und etwaigen Mitarbeitern bzw. Mitarbeitergruppen zugeteilt werden. Ereignisse und Ergebnisse (u. a. Bearbeitungs-, Transport- und Liegezeiten) während der automatischen Abarbeitung werden protokolliert und für weitere Analysen herangezogen. Der Workflow-Regelkreis erlaubt somit eine stetige Optimierung des Geschäftsprozesses respektive Workflows.

Die oben skizzierten Phasen sind nach Müller in die zwei Kategorien Geschäftsprozessmanagement und Workflow-Management zu unterteilen.⁵⁷ In der Kategorie des Geschäftsprozessmanagement spiegelt sich überwiegend die fachlich-konzeptionelle Ebene wieder. Sie beschäftigt sich mit der Modellierung, Analyse und Optimierung.⁵⁸ Die zweite Kategorie, das Workflow Management, konzentriert sich auf die Umsetzung und Steuerung der erstellten Modelle.⁵⁹ Sie ist der technischen Ebene zuzuordnen. Obwohl Müller darauf verweist, dass für beide Kategorien unterschiedliche Software-Werkzeuge bereitstehen⁶⁰ ist eine derartige Trennung nicht zwingend. So bieten heutige Workflow-Management-Systeme im Allgemeinen eine ganzheitliche Unterstützung des Workflow-Regelkreises, wie in nachfolgendem Unterabschnitt gezeigt wird.

2.3.2 Begriffsdefinition

Ein Workflow-Management-System ist ein Softwaresystem, welches die Aufgaben des Workflow Managements durch entsprechende Softwarekomponenten unterstützt.⁶¹ Trotz Bemühungen der Workflow Management Coalition (WfMC) eine Standardisierung und Vereinheitlichung verschiedener Begrifflichkeiten auf dem Gebiet des Workflow Management voranzutreiben wird u. a. der Begriff Workflow-Management-System nicht

⁵⁵ vgl. [Mü06, S. 28]

⁵⁶ ibidem vgl. [Mü06, S. 28]

⁵⁷ ibidem vgl. [Mü06, S. 28 f.]

⁵⁸ ibidem vgl. [Mü06, S. 29]

⁵⁹ ibidem vgl. [Mü06, S. 29]

⁶⁰ ibidem vgl. [Mü06, S. 29]

⁶¹ ibidem vgl. [Mü06, S. 11]

einheitlich verwendet. Beispielsweise ist in [Hag04] eine leicht verallgemeinernde und in [Gad12] eine in gleichem Maße detailliertere Form des Begriffes zu finden. Für die weiteren Betrachtungen wird sich ausschließlich auf die Definition der Workflow Management Coalition bezogen. Die Existenz weitere Definitionen soll an dieser Stelle lediglich erwähnt und bei näherem Interesse auf die Quellen verwiesen werden.⁶² Nach der Workflow Management Coalition handelt es sich bei einem Workflow Management Coalition um:

„A System that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where require, invoke the use of IT tools and applications“.⁶³

Ein Workflow-Management-System unterstützt demnach ganzheitlich den Workflow-Regelkreis. Von zentraler Bedeutung für eine automatische Unterstützung des Workflows ist eine vorhandene und für eine Maschine interpretierbare Prozessdefinition. Diese entspricht im Allgemeinen dem oben bereits besprochenen Workflow-Modell, weshalb die Begriffe auch synonym verwendet werden.⁶⁴

Unerwähnt bleibt, dass Workflows über Unternehmensgrenzen hinweg automatisiert werden und dabei verschiedene Workflow-Management-Systeme zum Einsatz kommen können. Um die Interoperabilität verschiedener Workflow-Management-Systeme zwischen Herstellern zu fördern, hat die Workflow Management Coalition bereits frühzeitig begonnen eine Referenzarchitektur für den Aufbau eines Workflow-Management-Systems zu entwickeln.⁶⁵

2.3.3 Referenzarchitektur der WfMC

In Übereinkunft mit den Zielen⁶⁶ der Workflow Management Coalition soll die Referenzarchitektur zu einer Abgrenzung und Systematisierung der Systemkomponenten von Workflow-Management-Systemen beitragen.⁶⁷ Die Workflow Management Coalition differenziert hierbei zwischen sechs verschiedenen Systemkomponenten:

- dem Workflow Enactment Service,

⁶² Beispiel für eine allgemeinerer [Hag04, S. 137] und eine detailliertere [Gad12, S. 228] Form des Begriffes Workflow-Management-System.

⁶³ [Coa94, S. 9]

⁶⁴ vgl. [Hag04, S. 30]

⁶⁵ Die aktuelle Referenzarchitektur stammt aus dem Jahr 1995.

⁶⁶ Die Ziele liegen in der Entwicklung und Verbreitung von Standards (bspw. standardisierte Interfaces, Datenaustauschformate und einheitliche Software-Terminologie) sowie eine Verbesserung der Interoperabilität, infolgedessen eine erhöhte Akzeptanz und Verbreitung von WfMS erreicht werden soll. vgl. [Hag04, S. 161]

⁶⁷ vgl. [Gad12, S. 231]

- den Process Definition Tools
- den Workflow Client Applications,
- den Invoked Applications,
- den Administration & Monitoring Tools und
- den Other Workflow Enactment Service(s).

Diese werden über Interfaces (dt. Schnittstellen) angesteuert, welche die Kommunikation zwischen den Systemkomponenten regulieren. Modellhaft wird das Zusammenspiel der Systemkomponenten in Abbildung 2.5 veranschaulicht.

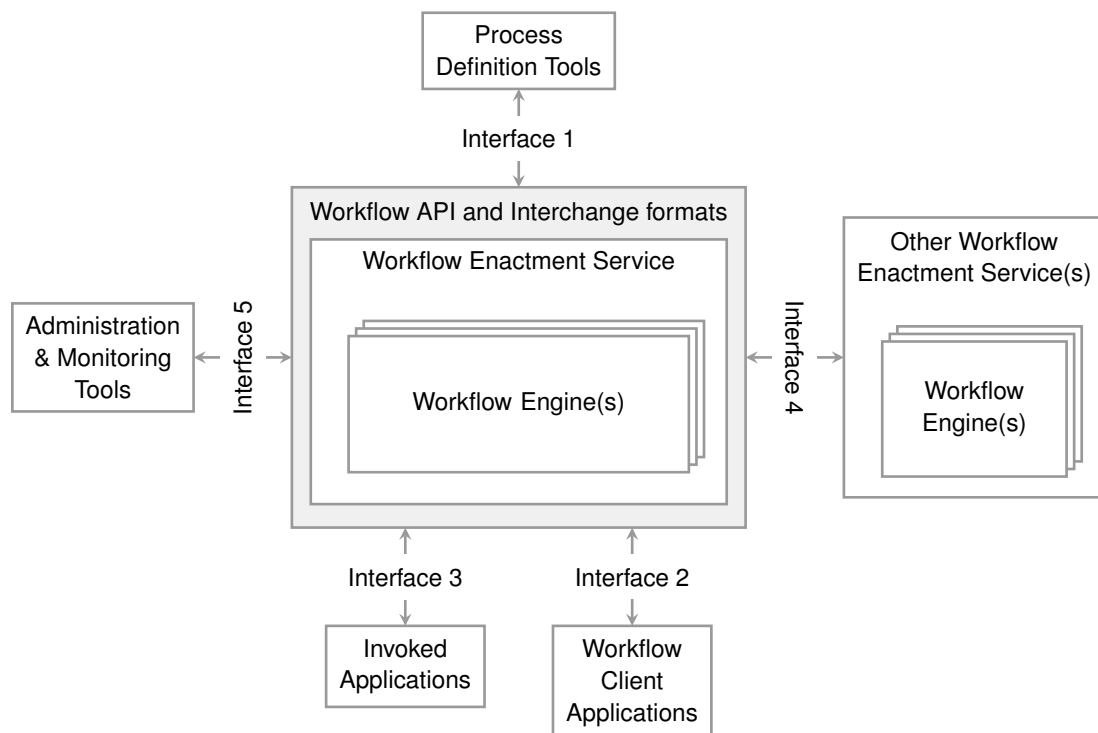


Abbildung 2.5: Referenzarchitektur der WfMC⁶⁸

Interface-Spezifikationen machen Systemkomponenten austauschbar. Ein Workflow-Management-System kann demnach aus Software-Produkten (Systemkomponenten) verschiedener Hersteller bestehen. Für ein besseres Verständnis über die Aufgaben und Funktionsweisen einzelner Systemkomponenten sowie Interfaces sollen diese im Folgenden kurz vorgestellt werden.

Workflow Enactment Service

Der im Zentrum stehende Workflow Enactment Service kann mehrere Workflow Engines enthalten. Er ist zuständig für die Erzeugung, Verwaltung und Ausführung von

⁶⁸ vgl. [Gad12, S. 232]

Workflow-Instanzen.⁶⁹ Zusätzliche Systemkomponenten sind über das Workflow Application Programming Interface (WAPI) anzubinden.⁷⁰ Für die Synchronisation in verteilten Workflow-Management-Systemen sind abhängig von der angebundenen Systemkomponente spezifische Protokolle und Austauschformate (engl. Interchange formats) zu unterstützen.

Process Definition Tools

Workflow-Modelle sind über die Process Definition Tools zu erstellen. Mit Ausnahme der Modellierung ist der konkrete Funktionsumfang stark herstellerspezifisch. Etwaige Erweiterungen bilden die Analyse, Modellbeschreibung und Dokumentation von Workflow-Modellen.⁷¹ Über das Interface 1 (Workflow Definition Interchange) erfolgt der Transfer der Workflow-Modelle zum Workflow Enactment Service,⁷² wo sie von mindestens einer Workflow Engine verarbeitet werden können. Um die Herstellerunabhängigkeit zu fördern, beschreibt die Workflow Management Coalition ein Metamodell, welches Mindestanforderungen an ein Workflow-Modell festlegt. Es definiert essentielle Objekte, Attribute sowie deren Beziehungen, die zwingend von einem Workflow-Modell zu unterstützen sind.⁷³ Eine Aufzählung aller zu einem Objekt gehörenden Attribute findet sich im Anhang A. Eine Übersicht der essentiellen Elemente ist Abbildung 2.6 zu entnehmen.

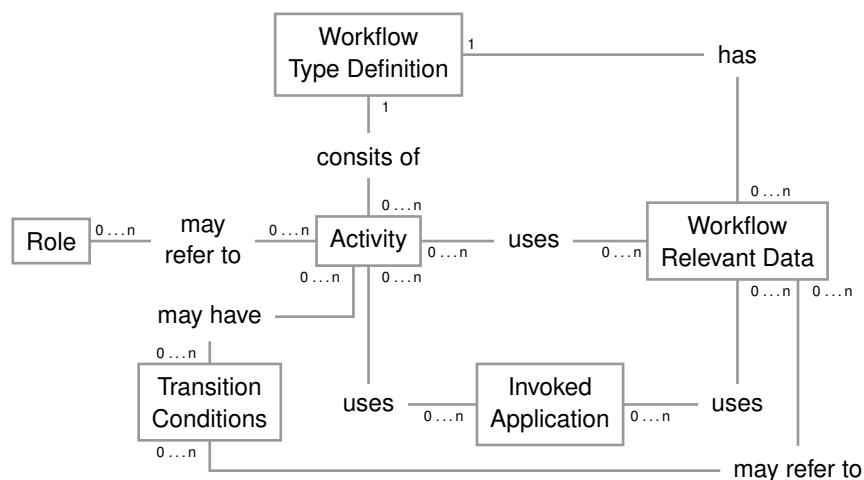


Abbildung 2.6: Workflow-Metamodell der WfMC⁷⁴

Ein Workflow-Modell besteht demnach aus einer Vielzahl an Aktivitäten, welche verschiedene Daten (Workflow Relevant Data) für die Ausführung verwenden. Die Daten

⁶⁹ vgl. [Coa95, S. 21]

⁷⁰ ibidem vgl. [Coa95, S. 21]

⁷¹ vgl. [Hag04, S. 163]

⁷² vgl. [Coa95, S. 28 f.]

⁷³ ibidem vgl. [Coa95, S. 29]

⁷⁴ ibidem vgl. [Coa95, S. 30]

können dem Modell beispielsweise als Eigenschaften, in Form von Variablen, hinzugefügt werden. Übergangsbedingungen (Transition Conditions) nutzen die angegebenen Daten sogleich für eine präzise Ablaufsteuerung. So kann beispielsweise anhand eines Schwellwertes, innerhalb der Übergangsbedingung sowie mit Hilfe der übergebenen Daten, eine korrekte Auswahl der Folgeaktivität vorgenommen werden. Die Zuständigkeit und Ausführbarkeit einer Aktivität regelt das Element Rolle. Eine Rolle repräsentiert in diesem Kontext beispielsweise eine Abteilung bzw. einen Benutzer. Jeder Benutzer der über eine Rolle einer Aktivität zugeordnet ist, verfügt über die entsprechende Berechtigung eben diese auszuführen. Über das Interface 3 ist es in diesem Zusammenhang möglich Aktivitäten innerhalb eigener Anwendungen (Invoked Application) auszuführen. Dabei ist es möglich die Workflow-Daten für die Ausführung miteinzubeziehen.

Neben einer Nennung von Mindestanforderungen bildet das Metamodell die Basis für die Erstellung einer Reihe von Austauschformaten (engl. Interchange formats) zwischen verschiedenen Herstellern.⁷⁵ Für die automatische Ausführung der Workflows muss das Workflow-Modell in BPEL bzw. XPD L vorliegen. Ein spezifisches Austauschformat muss daher gewährleisten, dass das Workflow-Modell weiterhin von der Workflow Engine verarbeitet werden kann.

Workflow Client Applications

Workflow Client Applications gestatten die Interaktion zwischen Benutzer und Workflow Enactment Service.⁷⁶ Allgemein findet die Interaktion über ein speziell dafür vorgesehene Benutzeroberfläche statt. Diese kann bereits im Workflow-Management-System enthalten oder eigens implementiert werden.⁷⁷ Als quasi Standard hat sich hierfür das Konzept der Worklist (dt. Arbeitsliste) durchgesetzt.⁷⁸ Die Liste beinhaltet sämtliche aktiven Aufgaben aller laufenden Workflows, die eine Benutzerinteraktion implizieren. Ein einzelner Eintrag wird als Workitem bezeichnet und kann verschiedenen Organisationseinheiten zugeordnet werden. Einträge in der Worklist müssen sich nicht auf ein Workflow-Management-System beschränken. Um die Menge verschiedener Worklists zu minimieren, ist eine vereinte Darstellung in einer gemeinsamen Worklist möglich und ratsam.⁷⁹ Über das Interface 2 (Workflow Client Application Interface) und den sich dahinter verbergenden, vom Enactment Service bereitgestellten WAPI-Funktionen erfolgt eine entsprechende Steuerung. Kernfunktionen bilden hierbei:⁸⁰

- der Verbindungsaufbau und -abbau,

⁷⁵ vgl. [Coa95, S. 29]

⁷⁶ ibidem vgl. [Coa95, S. 31]

⁷⁷ ibidem vgl. [Coa95, S. 31 f.]

⁷⁸ vgl. [Hag04, S. 163]

⁷⁹ vgl. [Coa95, S. 32 f.]

⁸⁰ ibidem vgl. [Coa95, S. 33 ff.]

- die Prozess- und Aufgabekontrolle,
- der Abruf von Prozessstatus und
- die Worklist-Manipulation.

Die Steuerung übernimmt die Workflow Engine, welche ihre Funktionen in Form von Dienstleistungen den Workflow Client Applications anbietet.⁸¹ Die Interaktion beginnt demnach stets auf Seite der Workflow Client Applications und wird von einem Nutzer ausgelöst. Beispielsweise ein Abruf aktueller Worklist-Einträge für einen Benutzer.

Invoked Applications

Wie Workflow Client Applications interagieren Invoked Applications mit einem Workflow Enactment Service.⁸² Sie unterscheiden sich von Workflow Client Applications dahingehend, dass sie die Ausführung einzelner Aufgaben und Work Items unterstützen und eine Initialisierung über das Workflow-Management-System erfolgt.⁸³ Als Work Item versteht man hierbei alle im Kontext einer Aktivität anfallenden Arbeiten innerhalb eines Workflows. Invoked Applications sind über das Interface 3 (Invoked Application Interface) anzusprechen. Für eine mühelose Anbindung von Fremdsystemen und Work Items fasst das Interface 3 verschiedene Standards zusammen.⁸⁴ Hierzu zählen u. a. Local Process Calls, ORB Calls, Remote Execution Calls und das Message Passing.⁸⁵ Invoked Applications können sich lokal zur Workflow Engine (z. B. deployed innerhalb eines Application Servers), auf dem selben System (z. B. Windows, Unix) oder auf einem separaten System im Netzwerk befinden.

Other Workflow Enactment Service(s)

Die Anbindung weiterer Workflow Engines (verschiedener Hersteller) ist über das Interface 4 (WAPI Interoperability Functions) zu realisieren. Die Workflow Management Coalition definiert 4 Modelle der Interoperabilität: a) Diskret zusammenhängend (verkettet), b) hierarchisch (verschachtelte Subprozesse), c) indiskret zusammenhängend (Peer-To-Peer) und d) parallel synchronisiert (über Synchronisationspunkt).⁸⁶ Abbildung 2.7 veranschaulicht die genannten Modellen noch einmal beispielhaft. Die vier Modelle zeigen stets zwei Workflows (Workflow 1 $\hat{=}$ ■ , Workflow 2 $\hat{=}$ □), welche jeweils von einer separaten Workflow-Engine (Workflow 1 \Rightarrow Workflow-Engine A, Workflow 2 \Rightarrow Workflow-Engine B) ausgeführt werden.

⁸¹ ibidem vgl. [Coa94, S. 41]

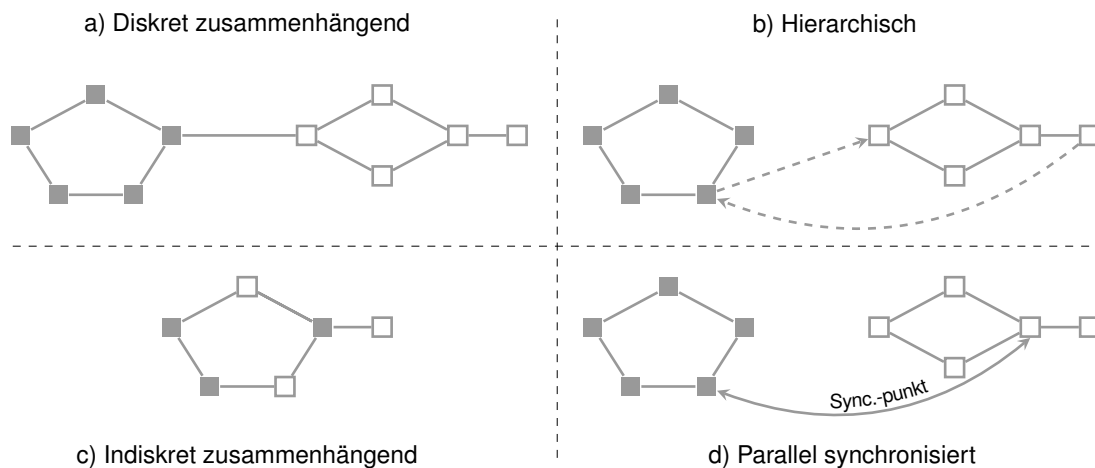
⁸² ibidem vgl. [Coa94, S. 41]

⁸³ vgl. [Coa94, S. 41]

⁸⁴ vgl. [Gad12, S. 231]

⁸⁵ vgl. [Coa95, S. 35 f.], Tabelle 1.

⁸⁶ ibidem vgl. [Coa95, S. 38 ff.]

Abbildung 2.7: Interoperabilitätsmodelle für WfMS⁸⁷

Für die Gewährleistung der Interoperabilität sind zwei wesentliche Aspekte relevant:⁸⁸

- Eine gleiche Interpretation der Workflow-Modelle bzw. eines Ausschnittes sowie
- Laufzeitsupport zum Austausch verschiedener Kontrollinformationen und für den Transfer von Workflow-Daten bzw. anwendungsrelevanter Daten zwischen verschiedenen Workflow Enactment Services.

Die Kommunikation erfolgt hierbei direkt über die WAPI-Funktionen oder über eine zwischengeschaltete Gateway-Anwendung.⁸⁹ Letzteres findet vor allem bei Workflow-Management-Systemen Einsatz, deren oben genannte Interoperabilitätsaspekte eine zu geringe Schnittmenge aufweisen. In einem solchen Fall führen Gateway-Anwendung ein Mapping verschiedener Objekte und Daten durch, wobei verschiedene Kommunikationsprotokolle Verwendung finden können.⁹⁰

Administration and Monitoring Tools

Administration and Monitoring Tools gestatten eine vollständige Statusüberwachung laufender Workflows.⁹¹ Simultan bieten sie eine umfassende Sammlung von Funktionen zur Administration unter besonderer Berücksichtigung der Aspekte Sicherheit, Kontrolle und Autorisierung.⁹² Wenngleich Administrations und Monitoring Tools integraler Bestandteil eines Workflow-Management-Systems sein können, ist eine gemeinsame Verwaltung einer Vielzahl an Workflow-Management-Systemen darüber hinaus möglich. Das Interface 5 (Administration & monitoring Interface) stellt diesen Sachverhalt

⁸⁷ ibidem vgl. [Coa95, S. 83 ff.], eigenen leicht abgeänderte Darstellung.

⁸⁸ ibidem vgl. [Coa95, S. 42]

⁸⁹ vgl. [Coa95, S. 41 ff.]

⁹⁰ ibidem vgl. [Coa95, S. 42]

⁹¹ ibidem vgl. [Coa95, S. 44]

⁹² ibidem vgl. [Coa95, S. 44]

sicher. Dahinter verbergen sich spezifizierte WAPI-Funktionen zu deren typischen Aufgaben u. a.:⁹³

- Benutzermanagement,
- Rollenmanagement,
- Ressourcenkontrolle und
- Prozessüberwachungsfunktionen zählen.

2.3.4 Architekturkonzepte für WfMS

Bisherige Ausführungen charakterisierten die verschiedenen Systemkomponenten eines Workflow-Management-Systems. Unbeantwortet bleibt die Frage nach dem konkreten Architekturkonzept. Wie fügen sich die Systemkomponenten in eine bestehende IT-Systemlandschaft ein? Hagen unterscheidet diesbezüglich zwischen form-based und engine-based Architekturen. Gadatsch präsentiert sogleich ein Client-Server-Modell für Workflow-Management-Systeme.

Form-based Architektur

Form-based Architekturen beruhen auf eine Wechselbeziehung zwischen einem Mailsystem und einer Menge benötigter Eingabemasken. Zu diesem Zweck verwaltet ein Messaging Service die Abarbeitungsreihenfolge vordefinierter Aufgaben.⁹⁴ Eingabemasken repräsentieren das User Interface und besitzen jederzeit Zugriff auf eine Datenbank, welche optional relevante Daten zu den etwaigen Aufgaben bereitstellt, wie in Abbildung 2.8 illustriert.⁹⁵

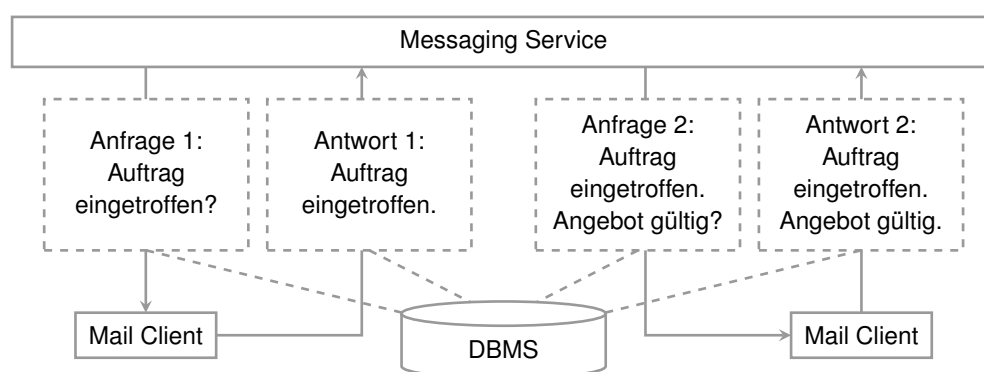


Abbildung 2.8: Form-based Architektur⁹⁶

⁹³ ibidem vgl. [Coa95, S. 44]

⁹⁴ vgl. [Hag04, S. 146]

⁹⁵ ibidem vgl. [Hag04, S. 146]

⁹⁶ ibidem [Hag04, S. 147]

Persistente Nachrichten informieren Aufgaben innerhalb des Messaging Service über die erfolgreiche Abarbeitung vorgelagerter Aufgaben.⁹⁷ Es findet kein direkter Datenaustausch zwischen einzelnen Aufgaben statt. Lokale persistente Warteschlangen sorgen sogleich für eine asynchrone Kommunikation zwischen einzelnen Aufgaben unabhängig der zugrundeliegenden Plattformen bzw. verwendeten Protokolle.⁹⁸ Hinterlegt der Messaging Service eine Aufgabe für einen Benutzer in eine nicht leere Warteschlange, kann dieser über die Abarbeitungsreihenfolge selbst entscheiden. Die Antwort erfolgt über das Mail-System und einem, für die entsprechende Aufgabe hinterlegten, Formular. Ein wesentlicher Nachteil von form-based Architekturen ist die fehlende Ereignisbehandlung. So gibt es beispielsweise keine Möglichkeit Aufgaben, welche über einen längeren Zeitraum in der lokalen Warteschlange des Benutzers liegen automatisch zu beantworten bzw. den Benutzer darauf hinzuweisen diese mit erhöhter Priorität zu behandeln.⁹⁹

Engine-based Architektur

Einen alternativen Ansatz verfolgt das Architekturkonzept der engine-based Architektur. Für die Ausführung der Workflow-Modelle ist der auf einem Server liegende Enactment Service zuständig.¹⁰⁰ Die Verwaltung der Workflow-Modelle sowie Zustandsänderungen der Workflow-Instanzen übernimmt eine zentrale Datenbank.¹⁰¹ Basis für die Entscheidung über einen notwendigen Aufruf einer Task Application (dt. Aufgaben-Anwendung) liefert das Workflow-Modell.¹⁰² Der Aufruf geschieht direkt oder optional über einen Task Controller. Dieser kann beispielsweise eine Validierung der vom Benutzer eingegebenen Daten vornehmen. Eine modellhafte Darstellung zeigt Abbildung 2.9.

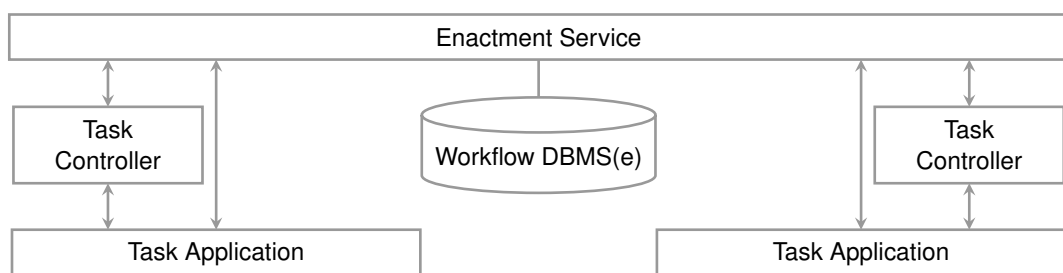


Abbildung 2.9: Engine-based Architektur¹⁰³

Ein wesentlicher Vorteil gegenüber der form-based Architektur ist die einfache Modellierung der Workflow-Prozesse mittels unabhängiger Modellierungswerkzeuge, welche

⁹⁷ ibidem vgl. [Hag04, S. 146]

⁹⁸ ibidem vgl. [Hag04, S. 146 f.]

⁹⁹ ibidem vgl. [Hag04, S. 147]

¹⁰⁰ vgl. [Hag04, S. 148]

¹⁰¹ ibidem vgl. [Hag04, S. 148]

¹⁰² ibidem vgl. [Hag04, S. 148]

¹⁰³ ibidem [Hag04, S. 148]

nach Fertigstellung direkt vom Enactment Service ausführbar sind.¹⁰⁴ Zeitgleich erlaubt die Speicherung u. a. von Zustandsänderungen eine nachträgliche Analyse des gesamten Workflows. Ein wesentlicher Nachteil gegenüber form-based Architekturen ist die Robustheit. So können Änderungen im Workflow-Modell zu Fehlfunktionen führen.¹⁰⁵

Client-Server-Schichtenmodell

Eine strengere Trennung der Verantwortlichkeiten unternimmt Gadatsch. Er beschreibt ein Client-Server-Schichtenmodell für Workflow-Management-Systeme. Ähnlich wie die engine-based Architektur erfolgt die Ausführung der Workflows auf einem Server unter Verwendung zusätzlicher Applikationen. Dabei wird von einer strikten Trennung zwischen dem prozesssteuernden Workflow-Management-System und den zur Unterstützung aufgerufenen Applikationen ausgegangen.¹⁰⁶ Auf dem Server befindet sich das Workflow-Management-System, während die Applikationssysteme anteilig auf Server und Client bereitgestellt werden können. Der Aufbau eines workflowbasierten Informationssystems, basierend auf einem Client-Server-Schichtenmodell, wird in Abbildung 2.10 exemplarisch dargestellt.

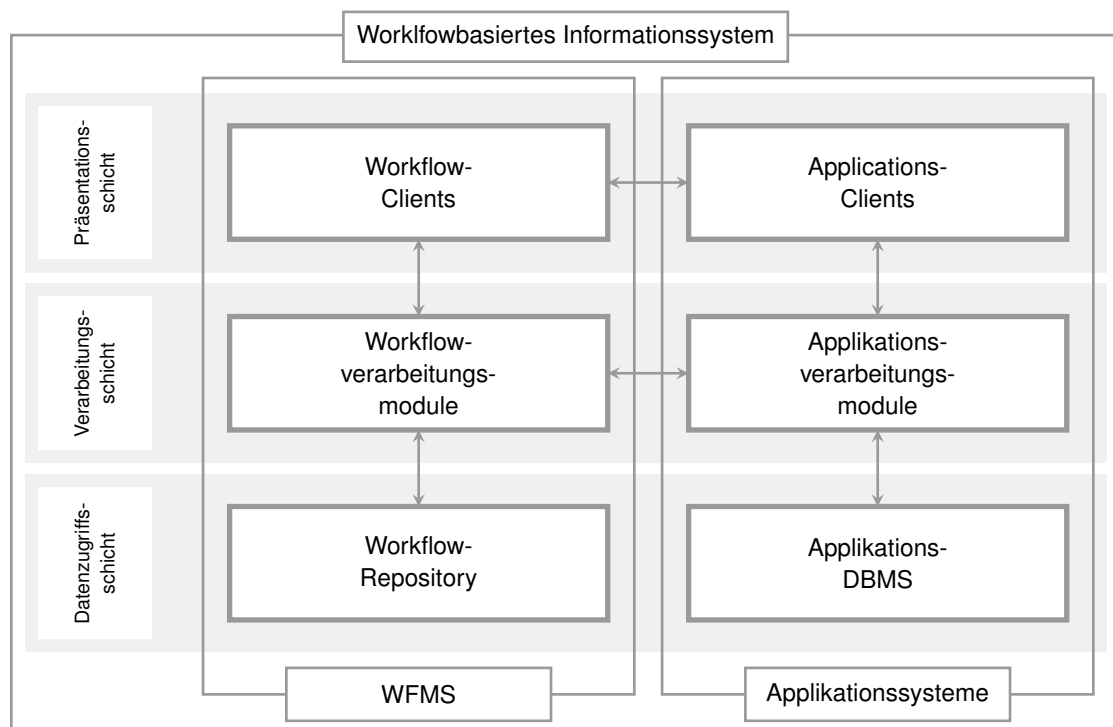


Abbildung 2.10: Client-Server-Schichtenmodell¹⁰⁷

Die Einteilung in eine Präsentations-, Verarbeitungs- und Datenübertragungsschicht

¹⁰⁴ ibidem vgl. [Hag04, S. 148]

¹⁰⁵ ibidem vgl. [Hag04, S. 148]

¹⁰⁶ vgl. [Gad12, S. 240]

¹⁰⁷ [Gad12, S. 241]

sorgt für eine Strukturierung einzelner Funktionalitäten innerhalb der beteiligten Systeme (Workflow-Management-System und Applikationssysteme).

Aus Nutzersicht stellt die Präsentationsschicht die erste Schicht dar. Sie beherbergt User-Interfaces (Dialogprogramme, zunehmend auch Web-Clients) zur Ausführung entsprechender Verarbeitungsmodule in der darunterliegenden Verarbeitungsschicht.¹⁰⁸ Gleichzeitig stellt sie eine Verbindung zwischen den Workflow- und Applikations-Clients bereit. Der Workflow-Client bildet hierbei die primäre Interaktionskomponente für den Benutzer. Applikations-Clients stellen eine Verbindung zu den Programmen der Applikationsmodule her, welche für die problembezogene Ausführung der Funktionen zur Verfügung stehen.¹⁰⁹ Ein Beispiel hierfür ist die Konfiguration einer Anwendung innerhalb des Applikations-Clients und einer anschließenden komplexen Berechnung des Applikationsmoduls. Im Rahmen der automatischen Workflow-Steuerung ist der Workflow-Client für den problembezogenen Aufruf und der Bereitstellung der für den Nutzer relevanten Applikations-Clients verantwortlich.¹¹⁰

In der Verarbeitungsschicht findet die eigentliche Prozesssteuerung und Verarbeitung definierter Workflow-Aktivitäten statt.¹¹¹ Die Funktionalitäten werden von Verarbeitungsmodulen bereitgestellt. Innerhalb eines Workflow-Management-Systems fallen hierunter beispielsweise Funktionen zur:¹¹²

- Administration (z. B. Benutzer-, Rechte- und Passwortverwaltung, etc),
- Simulation (z. B. Verifizierung von Korrektheit und Effizienz des Workflow-Modells),
- Analyse (z. B. von Ausführungszeiten von Workflows bzw. Aufgaben) und
- Steuerung von Workflows.

Fachfunktionen wie zum Beispiel zur Berechnung der Arbeitszeiten eines Mitarbeiters können über Applikationssysteme bereitgestellt werden. „Workflow-Verarbeitungsmodulen rufen hierzu ggf. Applikationsverarbeitungsmodulen auf“.¹¹³

Die Datenzugriffsschicht ist zuständig für die persistente Aufbewahrung und Bereitstellung aller zur erfolgreichen Abarbeitung der Workflow-Modelle notwendigen Informationen. Abhängig vom betrachteten System versteht man darunter u. a. Simulationsdaten, Analysedaten bzw. Applikationsdaten (z. B. Arbeitszeitsätze der Mitarbeiter).¹¹⁴

¹⁰⁸ ibidem vgl. [Gad12, S. 240]

¹⁰⁹ ibidem vgl. [Gad12, S. 240]

¹¹⁰ ibidem vgl. [Gad12, S. 240]

¹¹¹ vgl. [Gad12, S. 140]

¹¹² vgl. [Gad12, S. 243 ff.]

¹¹³ [Gad12, S. 240]

¹¹⁴ [Gad12, S. 241 ff.]

2.4 Integration von Workflow-Management-Systemen

Die Integration eines Workflow-Management-Systems in ein bestehendes ERP-System ist auf verschiedenen Integrationsarten und -stufen umsetzbar. Deren Besonderheiten es zwingend bei der späteren Konzeptentwicklung zur Integration in das bestehende ERP-System zu beachten gilt.

2.4.1 Integrationsarten von WfMSe

Bei der Integrationsart wird zwischen einer eigenständigen (engl. standalone) und eingebetteten (engl. embedded) Anwendung unterschieden. Eingebettet bedeutet in diesem Zusammenhang, dass eine Abbildung aller benötigten Funktionen des Workflow-Management-Systems über das ERP-System erfolgt. Die Konfiguration und Ausführung einzelner Aufgaben (z. B. starten von Prozessinstanzen bzw. eines User Task) wird über das ERP-System gesteuert. Dies führt im allgemeinen zu einem erhöhten Integrationsaufwand. Gleichzeitig kann es die Akzeptanz der späteren Benutzer erhöhen, da die Steuerung über bereits bekannte Bedienmuster stattfindet.

Das Gegenstück zur Einbettung eines Workflow-Management-Systems in ein bestehendes ERP-System ist die Verwendung des Workflow-Management-Systems als eigenständige Anwendung bzw. als Service. Dies verringert den Integrationsaufwand und erhöht die Ausfallsicherheit: das Workflow-Management-System ist von einem Ausfall des ERP-Systems nicht betroffen. Die Anwendung sendet seine Anfragen (z. B. die Ausführungen einer Prozessinstanz) an den Service des Workflow-Management-Systems, welches im Anschluss die Anfrage bearbeitet und das Ergebnis zurück an das ERP-System übermittelt.

2.4.2 Integrationsstufen für WfMSe

Die Integrationsstufen geben Aufschluss über den Automatisierungsgrad, bezogen auf die Unterstützung der Benutzer durch das Workflow-Management-System. Eine wesentliche Rolle spielt hierbei der Aufruf der richtigen Anwendung, zum richtigen Zeitpunkt. Nach der in Abschnitt 2.3.3 vorgestellten Referenzarchitektur unterscheidet man zwei Anwendungstypen: Workflow Client Applications und Invoked Applications. Workflow Client Applications setzen stets eine Nutzerinteraktion voraus. Invoked Applications können jedoch auch ohne Nutzerinteraktion auskommen. Die Integrationsstufen reichen vom einfachen manuellen Anwendungsaufwurf durch den Nutzer (Integrationsstufe 0) bis zur vollständigen Automatisierung ohne Nutzereingriffe (Integrationsstufe 4). Nicht alle Anwendungen können die höchste Integrationsstufe erreichen.¹¹⁵ Gründe hierfür können u. a. in der mangelnden Unterstützung durch das Workflow-Management-System

¹¹⁵ vgl. [Gad12, S. 248]

bzw. der zu integrierenden Anwendung liegen, beispielsweise bei Anwendungen die einen parametrisierten Aufruf nicht unterstützen.¹¹⁶ Die nachfolgende Betrachtung stellt die Integrationsstufen kurz vor und gibt einen Einblick in die Möglichkeiten der Anwendungsintegration.

Bei Integrationsstufe 0 (manuelle Ausführung) ist eine Anwendungsintegration nicht existent. Die Ausführung der Aufgaben ebenso wie ein eventueller Aufruf unterstützender Anwendungen geschieht manuell durch den Nutzer.¹¹⁷ Das Workflow-Management-System hat zu keinem Zeitpunkt Kontrolle und steht in keiner Verbindung zur aufgerufenen Anwendung. Es ist allein für die Steuerung des Workflows zuständig.¹¹⁸ Das Workflow-Management-System informiert den Nutzer über anstehende Aufgaben und der Nutzer informiert das Workflow-Management-System über deren Erledigung. Die Art und Weise der Abarbeitung bleibt dem Workflow-Management-System verborgen.

In Integrationsstufe 1 (Manuelle Ausführung mit Anwendungsoption) stellt das Workflow-Management-System dem Nutzer eine Auswahl an Anwendungen bereit, deren Verwendung nicht zwingend ist.¹¹⁹ Der Nutzer ist frei in seiner Entscheidung eine alternative Anwendung zu starten.¹²⁰ Bei Auswahl einer vom Workflow-Management-System vorgeschlagenen Anwendung liegt diese unter Kontrolle des Workflow-Management-Systems.¹²¹ Eine Datenintegration, d. h. eine Bereitstellung zusätzlicher Informationen anderer Anwendungen, beispielsweise über Parameter, ist nicht Bestandteil.

In Integrationsstufe 2 (Anwendungsaufruf) werden Anwendungen ausschließlich über das Workflow-Management-System zur Verfügung gestellt.¹²² Die Wahlmöglichkeit alternativer Anwendungen entfällt.¹²³ Die Suche aller notwendigen Informationen erfolgt manuell durch den Nutzer. Eine Datenintegration ist nicht vorhanden. Die Protokollierung von Bearbeitungszeiten durch das Workflow-Management-System ist möglich. Gespeicherte Protokolldaten bilden die Grundlage für eine zukünftige Optimierung des Workflows.

Anwendungen der Integrationsstufe 3 (Anwendungsparametrisierung) unterstützen zusätzlich einen parametrisierten Aufruf.¹²⁴ Durch die mögliche Parametrisierung ist die maximale Form der Datenintegration erreicht. Nicht alle Applikationen unterstützen diese Form des Aufrufs.¹²⁵ Die übergebenen Daten können beispielsweise zur Vervollständigung von Formularfeldern herangezogen und vom Nutzer im Rahmen der Abarbeitung

¹¹⁶ vgl. [Gad12, S. 248 f.]

¹¹⁷ vgl. [Gad12, S. 247]

¹¹⁸ vgl. [Gad12, S. 247]

¹¹⁹ vgl. [Gad12, S. 247]

¹²⁰ vgl. [Gad12, S. 247]

¹²¹ vgl. [Gad12, S. 247]

¹²² vgl. [Gad12, S. 247 f.]

¹²³ vgl. [Gad12, S. 247 f.]

¹²⁴ vgl. [Gad12, S. 247 f.]

¹²⁵ vgl. [Gad12, S. 248]

verwendet werden. Hierdurch ist eine bessere Unterstützung des Nutzers und somit eine beschleunigte Abarbeitung der Workflow-Aufgabe möglich.

Die Integrationsstufe 4 (Anwendungsautomatisierung) bietet eine vollautomatische Anwendungsunterstützung durch das Workflow-Management-System. Anwendungen dieser Stufe sind vollständig integriert und werden ohne Nutzerinteraktion gesteuert.¹²⁶ Zuständig für die Bereitstellung aller anwendungsrelevanter Daten ist das Workflow-Management-System.¹²⁷ Eine bedingte Datenintegration erfolgt wie in Integrationsstufe 3 anhand von Parametern.

Die Benutzerinteraktion nimmt demnach mit steigender Integrationsstufe ab, während die Anwendungskontrolle durch das Workflow-Management-System zunimmt. Dies führt dazu, dass der Benutzer weniger Zeit für die Informationsbeschaffung aufbringen muss. Das Workflow-Management-System kann die Informationen direkt aus der Datenbank lesen und der Anwendung übergeben. Der Benutzer übernimmt zunehmend eine Kontrollfunktion und kann seine Ressourcen anderen Tätigkeiten zuführen. Die Datenintegration ist besonders für die Integrationsstufen 3 und 4 von hoher Bedeutung, wodurch ein parametrisierter Anwendungsaufwurf realisierbar ist. In Tabelle 2.7 werden die einzelnen Stufen noch einmal zusammengefasst und anhand eines Beispiels erläutert.

Tabelle 2.7: Stufen der Anwendungsintegration¹²⁸

Integrationsstufe	Beschreibung	Beispiel
Stufe 0 Manuelle Ausführung	Workflows werden ohne Applikationsunterstützung ausgeführt.	Ermittlung des zuständigen Sachbearbeiters zu einer Kundenanfrage
Stufe 1 Manuelle Ausführung mit Applikationsoption	Workflows werden mit optionaler Applikationsunterstützung ausgeführt.	Angebotsprüfung. Der Beobachter kann ein vorgeschlagenes Tool wahlweise verwenden.
Stufe 2 Applikationsaufruf	Workflows werden computergestützt ausgeführt. Das WfMS wählt die geeignete Applikation aus und startet sie. Der Bearbeiter führt weitere Schritte durch.	Antwortbrief. Das WfMS fordert den Bearbeiter auf, einen Brief anzufertigen und startet das Textverarbeitungsprogramm.

¹²⁶ vgl. [Gad12, S. 249]

¹²⁷ vgl. [Gad12, S. 249]

Tabelle 2.7: Stufen der Anwendungsintegration (Fortsetzung)

Integrationsstufe	Beschreibung	Beispiel
Stufe 3 Applikations- parametrisierung	Workflows werden computerunterstützt ausgeführt. Das WfMS wählt die geeignete Applikation aus, startet sie und wählt ggf. Parameter. Der Bearbeiter führt weitere Schritte durch.	Anlegen eines Kundenauftrags. Das WfMS ruft die Transaktionen auf und übergibt relevante Daten (z. B. Kundennummer), so dass der Bearbeiter vorgelegte Eingabefelder vorfindet.
Stufe 4 Applikations- automatisierung	Workflows werden automatisch ohne Benutzerinteraktion ausgeführt.	Nach Lieferscheinerstellung wird vom WfMS der Workflow „Faktura erstellen“ angestoßen.

3 Beschreibung der Ausgangssituation

Für eine erfolgreiche Integration ist der theoretische Hintergrund über das zu integrierende System nicht ausreichend. Zusätzlich bedarf es einem konkreten Wissen über diverse technische wie fachliche Aspekte der beteiligten Systeme. Erst nach einer Erfassung aller relevanten Informationen ist der Konzeptentwurf für eine spätere Integration denkbar. Dieses Kapitel versucht daher alle relevanten Aspekte für die Integration aufzuzeigen. Dabei gilt es nicht jede einzelne Funktion der beteiligten Systeme ganzheitlich zu beschreiben. Vielmehr sollen die von der Integration betroffenen Stellen identifiziert und damit einhergehende Konsequenzen für den Konzeptentwurf abgeleitet werden. Den Abschluss bildet eine Aufstellung notwendiger funktionaler und nicht-funktionaler Anforderungen, welche im Konzept zwingend zu berücksichtigen sind.

3.1 ERP-System hscERP

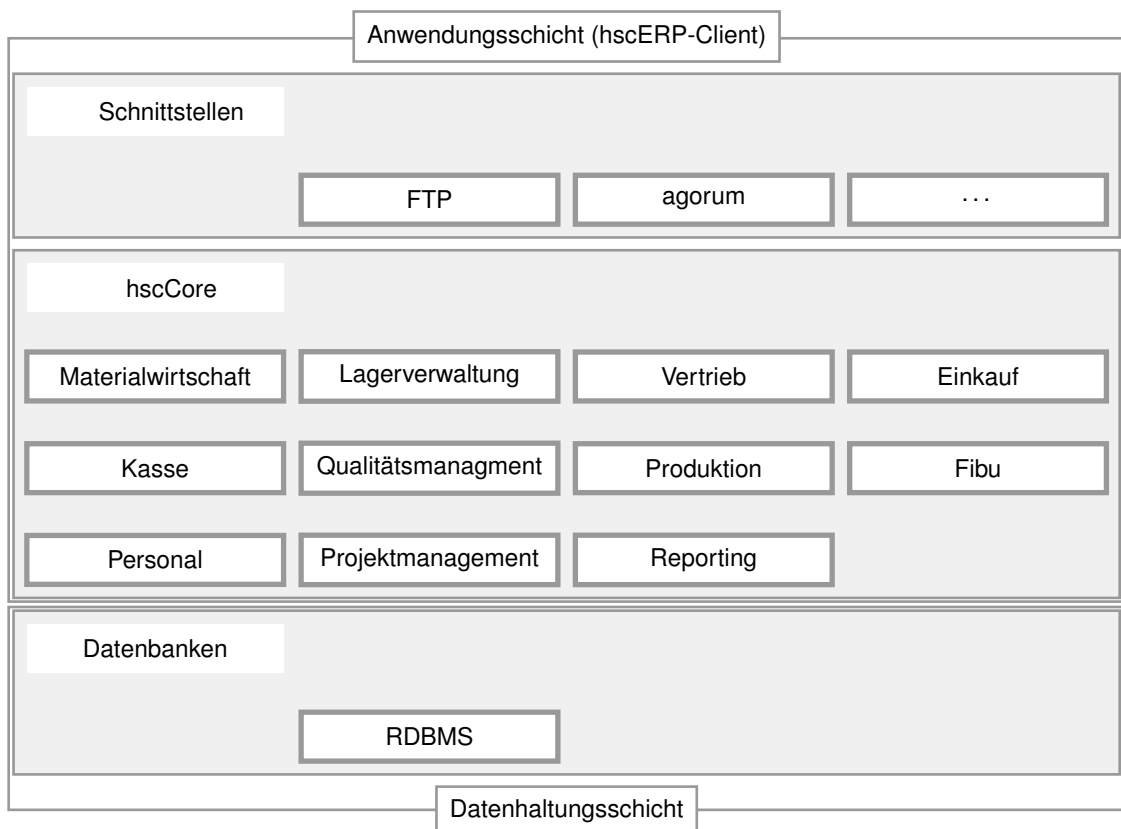


Abbildung 3.1: Schematischer Aufbau hscERP

Abbildung 3.1 zeigt den schematischen Aufbau des ERP-Systems *hscERP*, welches zu-

geschnitten ist auf die Bedürfnisse Kleiner und Mittlerer Unternehmen (KMU). Das Basissystem besteht aus einer zentralen Stammdatenverwaltung sowie modulübergreifenden Funktionen. Zu den modulübergreifenden Funktionen zählen unter anderem eine Benutzer- und Rechteverwaltung sowie ausführliche Logging-Funktionalitäten. Aufbauend auf dem Basissystem existiert ein funktionsbezogenes Modulportfolio wie in Abbildung 3.1 angedeutet. Daneben verfügt das *hscERP* über standardisierte Schnittstellen zum Datenaustausch mit Fremdsystemen (z. B. FTP, WebDAV, usw.) oder zur revisions-sicheren Dokumentenverwaltung, beispielsweise anhand des Dokumenten-Management-Systems (agorum® core), wie in Abbildung 3.1 skizziert.

Wie bereits einleitend erwähnt liegt das Hauptziel von ERP-Systemen in der Unterstützung unternehmensinterner Geschäftsprozesse. Das *hscERP* begünstigt sogleich eine unternehmensübergreifende Unterstützung von Geschäftsprozessen. Die Mandantenfähigkeit und die damit verbundene Möglichkeit zur Verwaltung von Geschäftsdaten verschiedener Unternehmen in einem zentralen RDBMS bilden hierfür die Basis.

3.1.1 Zugrundeliegende Architekturkonzepte

Das ERP-System *hscERP* liegt aktuell in einer Zwei-Schichten-Architektur vor, wie in Abbildung 3.2 stark abstrahiert dargestellt. Die Anwendungsschicht repräsentiert einen Fat-Client, der sowohl die interne Geschäftslogik einzelner Module, als auch das User Interface (UI, dt. Benutzer- bzw. Anwenderschnittstelle) für Benutzerinteraktionen beinhaltet. Ein lose gekoppeltes RDBMS sorgt auf der Datenhaltungsschicht für die zentrale Verwaltung aller Geschäftsdaten.

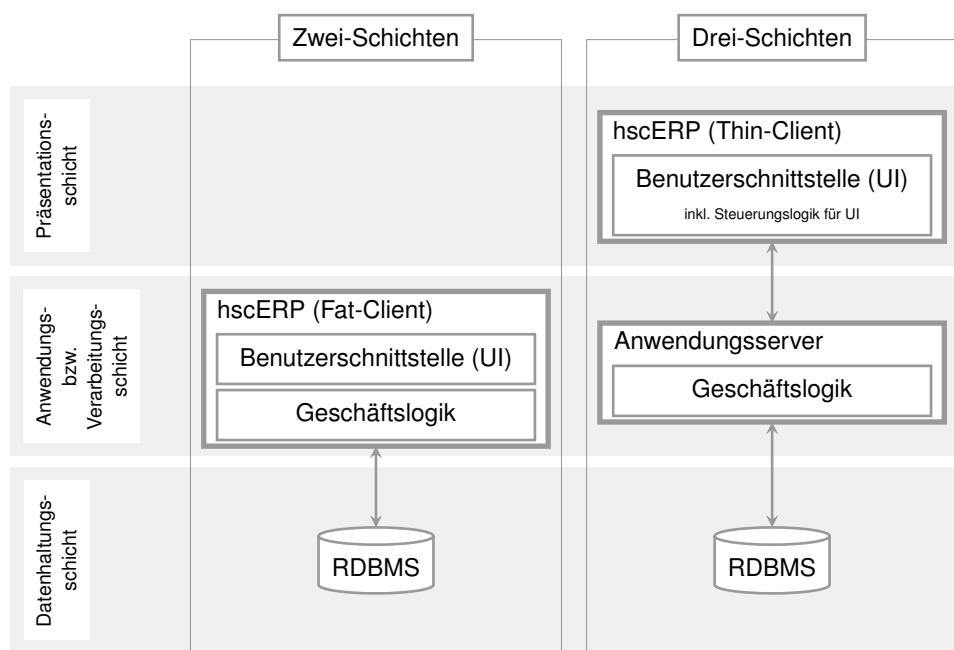


Abbildung 3.2: Portierung der Zwei-Schichten-Architektur auf eine Drei-Schichten-Architektur

Um die Flexibilität zu erhöhen, wird aktuelle eine Portierung auf eine Drei-Schichten-Architektur vorangetrieben. Durch das Abspalten von Geschäftslogik und UI entsteht eine dritte Schicht: Die Präsentationsschicht. Hierdurch konzentriert sich der Aufgabenbereich der ursprünglichen Anwendungsschicht verstärkt auf die Aufbereitung und Verarbeitung aller Geschäftsdaten, infolgedessen sie die Bezeichnung Verarbeitungsschicht trägt. Im Ergebnis führt dies zu einem Thin-Client, welcher lediglich das UI und die dafür notwendige Steuerungslogik enthält. Die Geschäftslogik wird auf einem Anwendungsserver (z. B. Wildfly-Application-Server) ausgelagert.

3.1.2 Interne Unterstützung der Architekturkonzepte

Im Rahmen der Portierung sind beide Architekturkonzepte zu unterstützen. Die sich daraus ergebende interne Struktur bezüglich des Klassenaufbaues zeigt Abbildung 3.3, welche auf Vorarbeiten von [Lei15b] zurückzuführen ist.

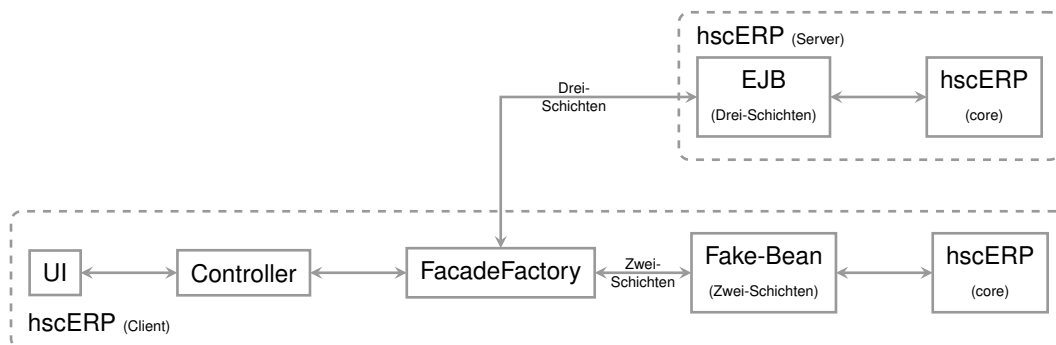


Abbildung 3.3: Interne Kommunikationsbeziehung zwischen Klassen

Das User Interface ist die Interaktionskomponente für den Anwender. Gut strukturiert zeigt sie dem Anwender alle wesentlichen Geschäftsdaten und erlaubt die Navigation durch selbige.

Der Controller enthält eine abstrakte Definition aller relevanten Operationen zur Anzeige und Manipulation der Geschäftsdaten. Als Beispiel sind hier die aus der Datenbankwelt allgemein bekannten CRUD-Operationen zum Erstellen (engl. Create), Lesen (engl. Read), Aktualisieren (engl. Update) und Löschen (engl. Delete) von Geschäftsdaten anzuführen. Der Controller selbst enthält keine konkrete Abbildung von Kontrollstrukturen zur Ausführung der entsprechenden Operation. Er definiert lediglich die angebotenen Schnittstellen und delegiert die Verantwortlichkeiten weiter an die FacadeFactory.

Die FacadeFactory bestimmt die verwendete Architektur. Eine parallele Nutzung beider Architekturkonzepte ist gegenwärtig nicht möglich. In Abhängigkeit der ausgewählten Architektur delegiert die FacadeFactory die vom Controller aufgerufene Funktionalität an die entsprechend dahinterliegenden Komponenten. Hierdurch ist die Abarbeitungs-ort der Funktionalität transparent für das UI. In der aktuellen Zwei-Schichten-Architektur

steuert eine Fake-Bean die grundlegenden Datenbankinteraktionen mit den Geschäftsdaten. In der zukünftigen Drei-Schichten-Architektur übernimmt diese Aufgabe eine Enterprise Java Bean (EJB). Dabei sind zusätzliche Faktoren, wie beispielsweise ein erweitertes Sitzungsmanagement zu berücksichtigen, welche u. a. für eine Übertragung der Geschäftsdaten im Netzwerk zwingend erforderlich ist.

Die *hscERP*-Komponenten beider Architekturvarianten enthalten die eigentliche Geschäftslogik, welche über die Fake-Bean bzw. EJB aufgerufen werden. Die Verwendung der *hscERP*-Komponenten erfolgt in Abhängigkeit der zugrunde gelegten Architektur auf dem Client bzw. Anwendungsserver.

3.1.3 Abbildung der Benutzer- und Rechteverwaltung

Das *hscERP* ist ein Mehrbenutzersystem. Für diesen Zweck verfügt es über eine eigene Benutzer- und Rechteverwaltung, welche im Kern aus acht Tabellen besteht:

- *dr_user* (Benutzer)
- *dr_user_priv* (Benutzerprivileg)
- *dr_usergroup* (Benutzergruppe)
- *dr_usergroup_member* (Benutzergruppenmitglied)
- *dr_usergroup_priv* (Benutzergruppenprivileg)
- *dr_activity* (Aktivität)
- *dr_activitygroup* (Aktivitätsgruppe)
- *dr_activitygroup_priv* (Aktivitätsgruppenprivileg)

Der Präfix *dr* dient der Prävention von Konflikten zwischen SQL-Schlüsselwörtern (z. B. *system*, *language*, *country*) und Tabellennamen. Für die nachfolgenden Betrachtungen ist dessen Existenz von geringer Bedeutung. Aus diesem Grund und für eine bessere Lesbarkeit gelten im weiteren Verlauf die in Klammern stehenden Begriffe.

Die Benutzertabelle enthält alle relevanten Benutzerinformationen. Hierzu zählen unter anderem Anmeldeinformationen wie Benutzername und das als Hash abgelegte Passwort ebenso wie die E-Mail-Adresse und eine Referenz auf die Anschrift des Benutzers. Alle Benutzer können Benutzergruppen zugeordnet werden. Die Herstellung einer Relation zwischen Benutzer und Benutzergruppe erfolgt über die Hilfstabelle Benutzergruppenmitglied.

Konkrete Rechte eines Benutzers bzw. einer Benutzergruppe werden über die Tabellen Aktivität, Aktivitätsgruppe und Aktivitätsgruppenprivileg gesteuert. Für die direkte Zuordnung von Benutzerrechten ist die Tabelle Benutzerprivileg entscheidend, für die Vergabe von Benutzergruppenrechten die Tabelle Benutzergruppenprivileg. Sie stellen die

Relationen zwischen den Tabellen Benutzer bzw. Benutzergruppe und Aktivitätsgruppe her. Eine Aktivitätsgruppe ist definiert als eine Menge von Aktivitäten. Eine Aktivität spiegelt eine Funktion wie z. B. das Benutzerlogin, das Öffnen eines Dialogfensters oder das grundlegende Betätigen eines Buttons wider. Eine Aktivität bildet noch kein Recht. Dies liegt unter anderem daran, dass Aktivitäten in verschiedenen Aktivitätsgruppen unterschiedlichen Einschränkungen unterliegen können. Die Rechtevergabe erfolgt über die Tabelle Aktivitätsgruppenprivileg. Sie gruppiert Aktivitäten zu Aktivitätsgruppen. Über Parameterspalten ist eine Definition entsprechender Privilegien (Rechte) für einzelne Aktivitäten innerhalb der Aktivitätsgruppe möglich. Beispielsweise, ob in einem eben geöffneten Dialog das Laden und Anzeigen (View) oder zusätzlich das Editieren und Löschen (Update) bzw. ausschließlich das Erzeugen (Create) von Geschäftsdaten erlaubt ist. Dies garantiert, dass Aktivitäten in verschiedenen Aktivitätsgruppen unterschiedliche Rechte zugewiesen werden können. Die Rechtevergabe findet in diesem Zusammenhang über parametrisierte Aktivitäten statt.

Abbildung 3.4 stellt das oben beschrieben noch einmal grafisch dar. Überflüssige Bezeichner und Attributnamen wurden vernachlässigt sowie bestehende zur besseren Lesbarkeit angepasst.

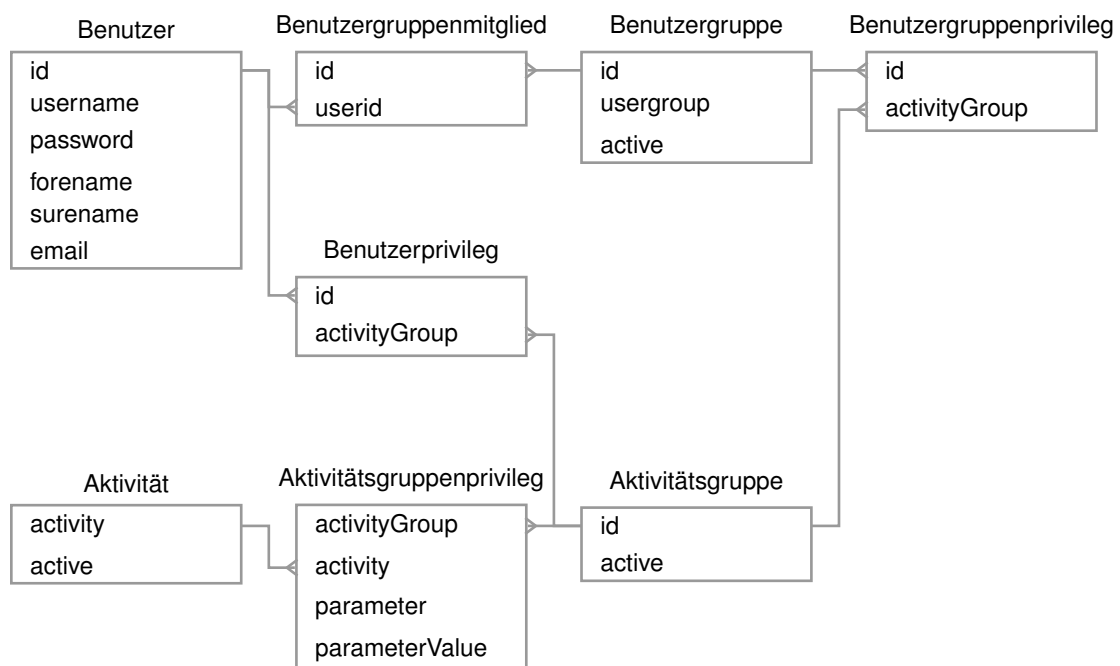


Abbildung 3.4: Übersicht Tabellen für Benutzer- und Rechteverwaltung

Eine Sonderstellung für die Benutzer- und Rechteverwaltung – und deshalb nicht mit in oben durchgeführter Aufzählung und Abbildung 3.4 aufgenommen – bildet die Tabelle web_user_roles. Sie weist einem Benutzer eine Web-Rolle zu, welche zum Beispiel für die Authentifizierung gegenüber einem Web- bzw. Anwendungsserver notwendig ist.

3.2 Workflow-Management-System jBPM

Unter Berücksichtigung der in Unterabschnitt 2.3.3 vorgestellten Referenzarchitektur der WfMC, gilt es im Folgenden das Workflow-Management-System *jBPM* im Detail vorzustellen und mögliche Integrationspunkte zu identifizieren.

3.2.1 Grundlegendes

Das Workflow-Management-System *jBPM* ist aktuell in Version 6.4 erhältlich.¹²⁹ Aufgrund der späten Veröffentlichung der aktuellen Version bezieht sich die vorliegende Arbeit auf Version 6.3. Eine Versionsaktualisierung während der Themenbearbeitung galt in Anbetracht des nicht zu unterschätzenden Konfigurationsaufwandes als nicht angemessen.

Als Open-Source-Software steht *jBPM* unter der Apache License 2.0.¹³⁰ Unter Berücksichtigung der Lizenzbestimmungen kann *jBPM* somit im kommerziellen Bereich frei verwendet werden.¹³¹ *jBPM* wurde für zwei Anwendergruppen entwickelt:¹³²

Fachanwender aus dem Bereich Business Process Management, welche für die Analyse und Modellierung der Geschäftsprozess verantwortlich sind, sowie
IT-Anwender (z. B. Softwareentwickler), welche die Fachanwender unterstützen den Geschäftsprozess zu einem ausführbaren Workflow weiterzuentwickeln.

3.2.2 Komponenten des WfMSs jBPM

jBPM verfügt über eine Vielzahl verschiedener Komponenten zur softwaretechnischen Unterstützung der oben genannten Anwendergruppen. Die einzelnen Komponenten sind in ihrer Verwendung unabhängig und lassen sich in Bezug auf ihrer Funktionalität in die vier Kategorien

- Runtime-Management,
- Modeling und Deployment,
- Execution sowie
- Reporting einteilen.

¹²⁹ Version 6.5 und 7.0 sind aktuell im Beta-Stadium und somit kurz vor einem 1. Release Candidate. Stand 24. Oktober 2016.

¹³⁰ vgl. [FG15, S. 21]

¹³¹ vgl. [Fou04]

¹³² vgl. [FG15, S. 21]

Aus Benutzersicht besonders von Interesse ist die KIE-Workbench.¹³³ Sie bildet das Bindeglied zwischen Benutzer und Workflow-Management-System. Als Web-App konzipiert ist sie über den Web-Browser aufrufbar und repräsentiert den Zugangspunkt zum Workflow-Management-System. Sie gewährt dem Benutzer Zugang zu den Funktionen des Runtime Managements, Reportings sowie zu den Modeling- und Deployment-Funktionen.

Runtime-Management-Komponente

Zu den Funktionen der Runtime-Management-Komponente gehören u. a. das Starten von Workflow-Modellen (KIE syn. Prozessdefinition) sowie das Stoppen und Abbrechen laufender Workflow-Instanzen (KIE syn. Prozessinstanzen). Von einem Workflow-Modell können zu Laufzeit mehrere Workflow-Instanzen gleichzeitig existieren. Anhand der integrierten Workflow-Verwaltung ist zu jeder Zeit einsehbar, welche Aufgabe eines Workflows sich aktuell in Bearbeitung befindet. Neben einer Workflow-Verwaltung beinhaltet das Runtime-Management Funktionen zur Verwaltung von User-Tasks. Innerhalb der KIE-Workbench geschieht dies über eine Worklist (KIE syn. Task-List).

Modeling- und Deployment-Komponente

Ebenfalls Bestandteil der KIE-Workbench ist der Prozessdesigner. Dieser erlaubt das Erstellen und Deployment von Workflow-Modellen. Die Erstellung der Workflow-Modelle erfolgt innerhalb eines grafischen Editors mittels BPMN. Zur Speicherung der Modelle nutzt *jBPM* BPMN 2.0,¹³⁴ welche neben der Darstellung, auch eine Ausführung der Modelle ermöglicht. Zu den weiteren Modellierungsfunktionen gehören

- der Data Modeler (zur Erstellung von Datenmodellen),
- der Form Modeler (zur Erstellung von Web-Formularen) und
- das Rules Authoring (zur Erstellung von Geschäftsregeln für regelbasierte Task).

Für die Verwaltung der verschiedenen Modelle sorgt ein Maven-Repository.¹³⁵ Es ist möglich mehrere Maven-Repositories anzulegen, deren zentrale Verwaltung anhand eines Guvnor-Repositories erfolgt. Dabei handelt es sich um eine Eigenentwicklung der Firma JBoss.¹³⁶ Das integrierte, verteilte Versionsverwaltungssystem Git ist für die Versionierung der abgelegten Modelle verantwortlich. Abgesehen von einer Modellierung mittels KIE-Workbench, besteht die Option zur Modellierung innerhalb der IDE Eclipse. Darüber modellierte BPMN-2.0-Modelle lassen sich über eine Importfunktion in die

¹³³ Das Akronym KIE steht für Knowledge Is Everything

¹³⁴ Die ausführliche Spezifikation ist unter <http://www.omg.org/spec/BPMN/2.0.2/PDF/> abrufbar.

¹³⁵ Angelegten Modellierungsprojekte haben die Maven-Projektstruktur eines Java-Projekts.

¹³⁶ Detaillierte Informationen bezüglich Guvnor sind auf der Projektseite <http://guvnor.jboss.org/> zu finden.

KIE-Workbench importieren.

Execution-Komponente

Hinter der Execution-Komponente verbergen sich die Core Engine, der Human Task Service sowie Schnittstellen (CDI, REST und JMS) für die Kommunikation mit Fremdsystemen.

Die Core Engine ist die eigentliche Komponente zur Ausführung der Workflow-Modelle. Basierend auf der Programmiersprache Java, ist sie in eigene Java-Anwendungen integrierbar. Hierfür stehen Java-Bibliotheken zur Einbindung in ein entsprechendes Java-Projekt bereit. Alternativ ist die Core Engine als Service nutzbar und über die KIE-Workbench bzw. über die mitgelieferte Remote API durch die eigene Anwendung ansprechbar. Im Kern besteht die Core Engine aus einem Semantik-Modul zur semantischen Überprüfung der übergebenen Workflow-Modelle sowie einer Prozess Engine zur Ausführung der überprüften Workflow-Modelle.¹³⁷ Für jeden gestartet Workflow erzeugt die Process Engine aus dem Workflow-Modell eine Workflow-Instanz. Die Workflow-Instanzen sind von außen über die anliegende API steuerbar. Abbildung 3.5 stellt diesen Sachverhalt grafisch dar. Gleichzeitig erfolgt über die API die Anbindung des Human Task Services.

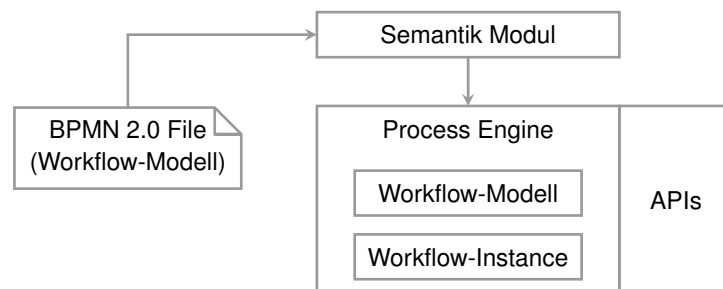


Abbildung 3.5: Module der Execution-Komponente¹³⁸

Der Human Task Service basiert auf der WS-Human-Task-Spezifikation¹³⁹ und ist zuständig für die Mensch-Maschine-Interaktion innerhalb des Workflows. Alle zu erledigenden User-Tasks eines Benutzers werden über den Human Task Service ausgeführt. Eine separate API erlaubt hierbei die Kommunikation zwischen Human Task Service und API der Process Engine sowie eine Integration in eigene Anwendungen.

Im Rahmen der Abarbeitung eines Workflows bzw. User-Tasks persistiert jBPM die ein-

¹³⁷ vgl. [MSA14, S. 31 f.]

¹³⁸ ibidem vgl. [MSA14, S. 31 ff.], eigene zusammenfassende Darstellung nach textlicher Beschreibung und grafischen Beispielen.

¹³⁹ Bietet u. a. eine XML-Syntax, um Aufgaben und Benachrichtigungen beschreiben zu können. Für eine umfassende Beschreibung der Spezifikation sei auf <http://docs.oasis-open.org/ns/bpml4people/ws-humantask/200803> verwiesen.

nehmbaren Status sowie Statuswechsel in einer Datenbank. Die erhobenen Daten bilden hierbei die Grundlage für spätere Analysen, beispielsweise zur Überprüfung der Bearbeitungszeiten.

Reporting-Komponente

Die Reporting-Komponente stellt Werkzeuge für das Business Activity Monitoring (BAM) bereit. Über den integrierten DashBuilder lassen sich diesbezüglich innerhalb der KIE-Workbench durch Ziehen und Ablegen einfache Reporte erstellen. Als Datenbasis dienen die von der Execution-Komponente persistierten Daten der Workflow- und Task-Abarbeitung.

3.2.3 Komponenten von jBPM unter Berücksichtigung des WfMC-Referenzmodells

Alle beschriebenen Komponenten lassen sich den definierten Interface-Typen der Workflow Management Coalition zuordnen. Eine klare Trennung ist im Fall von *jBPM* jedoch nicht gegeben. Dies lässt sich just auf die Vereinigung mehrerer Funktionalitäten innerhalb der KIE-Workbench zurückführen, welche sich gegenüber dem Benutzer als eine einzige Komponente präsentiert. Die Vereinigung stellt hierbei keinen Bruch bezüglich den Zielen der Workflow Management Coalition dar. Von der Workflow Management Coalition werden lediglich WAPI-Funktionen und deren Umsetzung spezifiziert, welche über die verschiedenen Interface-Typen gebündelt anzubieten sind. Unter Nutzung dieser Funktionen sind anschließend die oben beschriebenen Komponenten zu entwickeln. Eine Bereitstellung aller Komponenten innerhalb der KIE-Workbench in Form einer Web-App ist somit kein Widerspruch. Um die Austauschbarkeit der Komponenten zu gewährleisten, gilt es hierbei lediglich zu beachten die Funktionen unabhängig der darauf aufbauenden Komponente zu implementieren. Diesem Sachverhalt kommt *jBPM* durch eine angebotene Remote API nach. So bietet *jBPM*, neben den bereits erwähnten REST- und JMS-Schnittstellen, eine ausführliche und gut dokumentierte Java-API, welche die REST- bzw. JMS-Aufrufe zur Nutzung spezifischer Interface-Funktionen transparent gestaltet. Gleichzeitig erlaubt sie die Erstellung eigener Java-Anwendungen, in denen sich unter anderem verschiedene Funktionen der KIE-Workbench abbilden lassen.

3.2.4 Workflow-Modellierung und -Ausführung in jBPM

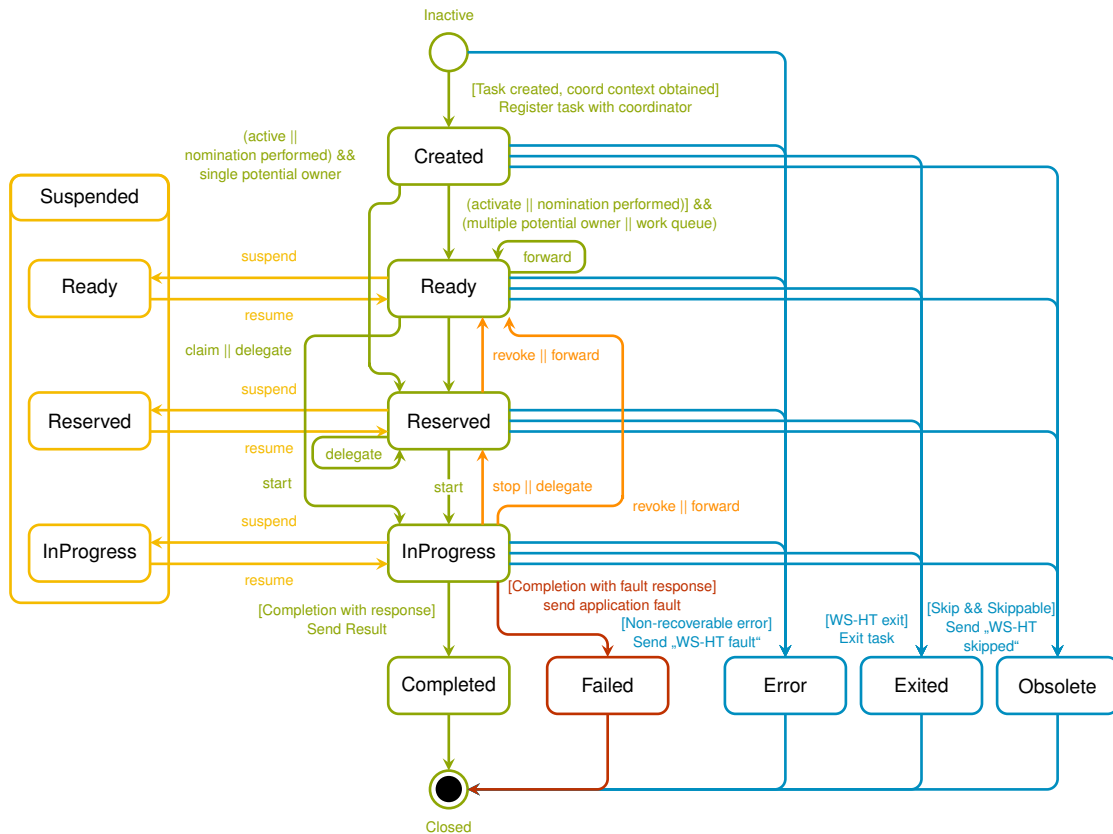
Die Modellierung der Workflow-Modelle erfolgt in *jBPM* standardmäßig über die KIE-Workbench. Hierfür ist zunächst ein neues Projekt anzulegen. Für die interne Verwaltung als Maven-Projekt ist eine Group-Id, Artifact-Id, Version-Id sowie der Projektna-

me anzugeben. Nach Anlegen eines Maven-Projekts öffnet sich der integrierte BPMN-Editor zur Erstellung von Workflow-Modellen. Zur eindeutigen Identifikation eines Workflow-Modells ist für jedes Modell eine alphanumerische *ProzessId* zu hinterlegen, welche sich aus dem Projekt- und Workflow-Namen (`projektname.workflowname`) zusammensetzt. Wie bereits einleitend erwähnt, sind für eine Automatisierung der Workflow-Modelle zusätzliche Informationen erforderlich. Im Hinblick darauf sind innerhalb des Workflow-Modells Variablen zu definieren, welche global dem gesamten Workflow bzw. lokal einzelnen Aktivitäten zugeordnet werden können. *jBPM* unterstützt in diesem Zusammenhang die aus der Java-Welt bekannten primitiven Datentypen (`char`, `byte`, `short`, `long`, `float`, `double`, `boolean`) sowie den Datentyp `String`. Der Geltungsbereich der Variablen ist hierbei abhängig von der Zuordnung. Globale Variablen sind innerhalb eines Workflows von allen Aktivitäten nutz- und veränderbar, lokale Variablen sind hingegen nur innerhalb einer Aktivität verwendbar. Da globale Variablen von allen Aktivitäten nutzbar sind, besteht die Möglichkeit eventuelle Ergebnisse aus abgearbeiteten Aktivitäten in globalen Variablen zu speichern und sie somit Folgeaktivitäten bereitzustellen.

Bei der Ausführung eines Workflows kann dieser verschiedene Status einnehmen. Hierzu zählen: *InProgress*, *Completed* und *Aborted*. Ein Abbruch einer Workflow-Instanz ist nach dem Start jederzeit manuell möglich.

3.2.5 Human Task in jBPM

Einen wesentlichen Schwerpunkt bei der Workflow-Abarbeitung repräsentiert die Problematik der Mensch-Maschine-Interaktion. Sie soll den Benutzer in den Mittelpunkt rücken, indem sie diesen bei der Bearbeitung anfallender Aufgaben optimal unterstützt. BPMN bietet in diesem Kontext die beiden Konzepte des User- bzw. Manuel-Tasks. Ein anstehender User-Task unterstützt den Benutzer hierbei direkt bei der Ausführung einer durchzuführenden Aufgabe. Hierzu zählt beispielsweise die Abfrage bzw. Bereitstellung erforderlicher Daten für die weitere Workflow-Abarbeitung. Ein Manual-Task hingegen ist als eine Art ToDo-Liste anzusehen, welche den Benutzer lediglich auf eine neue Aufgabe hinweist. Für eine revisionssichere Archivierung analoger Dokumente sind diese beispielsweise zunächst manuell von einem Benutzer zu digitalisieren. Auf die Ausführung dieser Aktivität hat das Workflow-Management-System keinerlei Einfluss. Während ein laufender Workflow beim Erreichen eines User-Tasks stoppt und bis zur erfolgreichen Abarbeitung an dieser Stelle verharret, läuft der Workflow beim Erreichen eines Manuel-Tasks weiter. Um dem Workflow-Management-System mitzuteilen, dass ein User-Task z. B. erzeugt, bearbeitet bzw. abgeschlossen ist, erfordert es eine Verwaltungskomponente. Diese wird innerhalb *jBPM* als Human Task Service bezeichnet. Eine eigenständige API bietet hierbei Zugriff für die Process Engine sowie Fremdsysteme. Eine ausführliche Darstellung der Status sowie Statusübergänge zeigt Abbildung 3.6 und wird allgemein als Human-Task-Lebenszyklus bezeichnet.

Abbildung 3.6: Human-Task-Lebenszyklus¹⁴⁰

Damit ein Human-Task von einem Benutzer ausführbar ist, muss diesem bei der Modellierung eine Benutzergruppe bzw. ein Benutzer zugewiesen werden. Fehlt eine Benutzerzuordnung ist ein Koordinator (Business-Administrator der KIE-Workbench) für die Zuordnung zur Laufzeit erforderlich, welcher dem Human-Task eine Benutzergruppe zuweisen kann. Der Human-Task befindet sich zu diesem Zeitpunkt im Status *Created*. Erfolgt eine Benutzer- bzw. Benutzergruppenzuordnung während der Modellierung, wird der Status *Created* übersprungen und direkt im Status *Ready* (Benutzergruppe) bzw. *Reserved* (Benutzer) fortgefahren. Ein Human-Task im Status *Ready* ist von einem Benutzer zu reservieren, wonach er sich im Status *Reserved* befindet und andere potenziell berechnete Benutzer der selben Benutzergruppe über keinen Zugriff mehr auf diesen verfügen. Die Reservierung eines Human-Task macht dessen Benutzer dabei zu dessen Besitzer. Gleichzeitig ist es möglich einen Human-Task im Status *Ready* durch einen Benutzer der selben Benutzergruppe an einen alternativen Benutzer zu delegieren, wonach er sich im Status *InProgress* befindet. Human-Tasks im Status *Reserved* bzw. *InProgress* können sogleich, wie in Abbildung 3.6 anhand der orangenen Pfeile dargestellt, zurück in die Status *Ready* bzw. *Reserved* gebracht werden, was einer Neuzuweisung des Human-Tasks bzw. einem Abbruch der Bearbeitung gleicht. Befindet sich ein Human-Task im Status *InProgress* wird dieser zum aktuellen Zeitpunkt von einem Benutzer ausgeführt, d. h. zu diesem Zeitpunkt findet die eigentliche Bearbeitung statt.

¹⁴⁰ vgl. [MSA14, S. 150], eigene leicht abgeänderte Darstellung.

Schlägt die Bearbeitung fehl und existiert eine Fehlerbehandlung in Form einer definierten Exception, wechselt er in den Status *Failed* und ist somit beendet. Ein Fehler in der Bearbeitung kann beispielsweise durch die Eingabe falscher bzw. Nichteingabe erforderlicher Daten auftreten. Im Falle einer unbehandelten Exception gelangt der Human-Task in den Status *Error*. Ist die Abarbeitung des Human-Tasks erfolgreich, gerät er in den Status *Completed* und ist somit abgeschlossen. Ein nicht abgeschlossener Human-Taks kann aus den Status *Created*, *Ready*, *Reserved* oder *InProgress* durch einen Wechsel in die Status *Exited* bzw. *Obsolete* vorzeitig beendet werden. Ein Wechsel in den Status *Exited* signalisiert in diesem Zusammenhang, dass die angeforderte Anwendung nicht länger an einem Ergebnis interessiert ist. Wurde der aktuell laufende Human-Task vom Benutzer (Besitzer) übersprungen, führt dies zu einer Statusänderung in den Status *Obsolete*. Ist die Abarbeitung kurzzeitig zu unterbrechen, beispielsweise aufgrund fehlender Informationen eines parallellaufenden Tasks, findet ein Wechsel in den Status *Suspended* statt.

3.2.6 Überblick relevante API-Elemente

Wie bereits oben erwähnt existieren für die Kommunikation mit Fremdsystemen verschiedene Schnittstellen (REST, JMS, CDI). Besonders interessant für die später Integration ist hierbei die Remote-API von *jBPM*. Diese erlaubt eine einfache Nutzung der angebotenen Schnittstellen über eine JAVA-API, welche die REST-Aufrufe innerhalb einer Vielzahl von Interfaces, Klassen und Methoden transparent gestaltet. Zu den wesentlichen API-Element für die Ausführung von Workflow-Modellen und Human Tasks gehören die Interface-Klassen:

- `RuntimeEngine`
- `KieSession`
- `TaskService` sowie
- `TaskSummary`.

Das `RuntimeEngine` Interface

Das `RuntimeEngine`-Objekt gestattet den Zugriff auf die Workflow Engine und den Human Task Service. Der Remote-Zugriffe über die REST- bzw. JMS-Schnittstelle erfolgt in diesem Zusammenhang über die abstrakte Klasse `RemoteRuntimeEngineFactory`. Diese stellt zwei Methoden für den Remote-Zugriff auf die `RuntimeEngine` bereit. Hierunter fallen die Methoden:

- `newJmsBuilder()` : `RuntimeEngine` sowie
- `newRestBuilder()` : `RuntimeEngine`.

In Abhängigkeit der gewählten Methode erfolgt die Kommunikation zwischen Client und Workflow-Management-System über das REST- bzw. JMS-Protokoll. Die RuntimeEngine ist hierbei als Laufzeitumgebung für Workflow-Instanzen anzusehen und entspricht dem oben beschriebenen Workflow-Enactment Service. Der in Listing 3.1 abgebildete Quellcodeausschnitt zeigt beispielhaft die Initialisierung der Remote Runtime Engine über REST.

Listing 3.1: Beispielhafte Initialisierung der RuntimeEngine über REST

```
1 RuntimeEngine runtimeEngine = RemoteRuntimeEngineFactory.newRestBuilder()  
2   .addTimeout(20) // In Sekunden. Wichtig für langsame Verbindungen!  
3   .addDeploymentId("de.hsc:testclient:1.0")  
4   .addUrl(new URL("http://localhost:8080/jbpm-console/"))  
5   .addUserName("admin")  
6   .addPassword("1234567890")  
7   .build();
```

Die *deploymentId* ist eine durch Doppelpunkt getrennte eindeutige Identifikationsnummer eines Maven-Projektes. Sie bezieht sich auf die Angaben der Group-Id (de.hsc), Artefact-Id (testclient) und Version (1.0) bei der Erstellung eines Maven-Projektes. Die URL verweist auf den Application-Server, welcher zuständig ist für die Ausführung und Verwaltung der KIE-Workbench und Workflow Engine. Nach erfolgreicher Initialisierung bietet das RuntimeEngine-Objekt zwei Methoden für den Zugang der KieSession und dem TaskService. Diese sind über zwei Getter-Methoden

- `getKieSession()` : KieSession und
- `getTaskService()` : TaskService abrufbar.

Das KieSession Interface

Das KieSession-Objekt besteht aus verschiedenen Interface-Klassen, welche die Ausführung und Verwaltung von Workflow-Instanzen erlauben. Zu den wesentlichen Methoden gehören:

- `getProcessInstance(String processId)` : ProcessInstance
- `getProcessInstances()` : Collection<ProcessInstance>
- `startProcess(String processId)` : ProcessInstance
- `startProcess(String processId, Map<String, Object> parameters)`
 : ProcessInstance
- `abortProcessInstance(long processInstanceId)` : ProcessInstance

Somit ist das KieSession-Objekt als zentrale Anlaufstelle zum Starten und Abbrechen von Workflow-Instanzen zu sehen. Gleichzeitig gestattet es eine Abfrage aller sich zur

Laufzeit in Ausführung befindlichen Workflow-Instanzen und gibt das entsprechende Ergebnis (in Form eines Collection-Objekts) zurück.

Das TaskService Interface

Das TaskService-Objekt gewährt Zugriff auf laufende User-Tasks und ermöglicht deren Verwaltung. Hierunter zählt primär die Statusänderung aktiver User-Task. Folgende Liste stellt einen Auszug wesentlicher Methoden zur Verwaltung von User-Tasks dar:

- `claim(long taskId, String userId) : void`
- `complete(long taskId, String userId, Map<String, Object> data) : void`
- `delegate(long taskId, String userId, String targetUserId) : void`
- `getTaskContent(long taskId) : Map<String, Object>`
- `getTaskById(long taskId) : Task`
- `getTasksAssignedAsPotentialOwner(String userId, String language) : List<TaskSummary>`
- `getTasksOwned(String userId, String language) : List<TaskSummary>`
- `getTasksOwnedByStatus(String userId, List<Status> status, String language) : List<TaskSummary>`
- `release(long taskId, String userId) : void`
- `resume(long taskId, String userId) : void`
- `skip(long taskId, String userId) : void`
- `start(long taskId, String userId) : void`
- `stop(long taskId, String userId) : void`

Bezogen auf den in Unterabschnitt 3.2.5 vorgestellten Human-Task-Lebenszyklus befindet sich ein gestarteter User-Task stets im Status *Ready* bzw. *Reserved*, da beim Start eine explizite Angabe des zuständigen Benutzers erforderlich ist. Anhand der über das TaskService-Objekt bereitgestellten Methoden lässt sich der in Unterabschnitt 3.2.5 beschriebene Human-Task-Lebenszyklus weitestgehend abbilden. Parallel stehen über das KieSession-Objekt Getter-Methoden bereit, welche anhand impliziter Suchkriterien eine Liste aktiver User-Tasks abfragbar ist. Als Suchkriterium dient hierbei beispielsweise der aktuelle Besitzer oder Status.

Das TaskSummary Interface

Das TaskSummary-Objekt liefert eine Übersicht aller abgearbeiteten und laufenden User-Tasks. Es stellt Logging-Informationen bereit, welche unter anderem den aktuellen Sta-

tus eines User-Tasks widerspiegeln. Die Kerninformationen sind über die folgenden Methoden abrufbar:

- `getId() : Long`
- `getName() : String`
- `getActualOwner() : User`
- `getCreatedBy() : User`
- `getDescription() : String`
- `getPotentialOwners() : List<String>`
- `getProcessSessionId() : Long`
- `getStatus() : Status`
- `getSubject() : String`
- `isQuickTaskSummary() : Boolean`
- `isSkipable() : Boolean`

Für das oben beschriebene `TaskService`-Objekt ist in diesem Zusammenhang besonders die darüber bereitgestellte `Id`-Information von Interesse. Diese stellt die im `TaskService`-Objekt benötigte `TaskId` dar. Erst hierdurch ist eine Verwaltung von User-Tasks möglich.

3.3 Anforderungen an das Integrationskonzept

Die bisherigen Abschnitte beschäftigten sich mit der Analyse des bestehenden ERP-Systems *hscERP* sowie dem zu integrierenden Workflow-Management-System *jBPM*. Den Abschluss dieser Analyse bildet die Definition allgemeiner sowie funktionaler und nichtfunktionaler Anforderungen. Diese bilden die Grundlage für das im Anschluss dieses Kapitels folgende verfeinerte Integrationskonzept.

3.3.1 Allgemeine Anforderungen

Die Integration des Workflow-Management-Systems erfolgt in Version 7 der Programmiersprache Java. Als Application-Server kommt ein WildFly-Application-Server in Version 9 zum Einsatz. Dieser ist verantwortlich für die Ausführung und Bereitstellung des Workflow-Management-Systems *jBPM* sowie der Geschäftslogik des *hscERPs*. Die Integration des Workflow-Management-Systems erfolgt als separates Modul innerhalb des ERP-Systems *hscERP*. Die Kommunikation zwischen *hscERP* und dem Workflow-Management-System *jBPM* erfolgt über das standardisierte REST-Protokoll.

3.3.2 Funktionale Anforderungen

Funktionale Anforderungen legen fest, welche Funktionen nach der Integration vom System übernommen und in Teilen dem Benutzer bereitgestellt werden. Nachfolgend findet diesbezüglich eine Beschreibung aller im Rahmen der angestrebten Integration ausgemachten funktionalen Anforderungen statt.

Bereich Application-Servers

Die Ausführung des Workflow-Management-Systems erfolgt innerhalb eines Application-Servers. Diesbezüglich muss es eine Möglichkeit geben Verbindungsinformationen im ERP-System zu hinterlegen, um eine Kommunikation zwischen ERP- und Workflow-Management-System zu gewährleisten. Zu den benötigten Informationen zählen in diesem Zusammenhang der Hostname sowie der verwendete Port. Diese Funktion ist ausschließlich für Systemadministratoren des Anwenderunternehmens zugänglich.

Bereich Workflow-Modelle

Die Verwaltung der Workflow-Modelle ist innerhalb der KIE-Workbench durchzuführen. Modellierung sowie Deployment der Workflow-Modelle für Anwenderunternehmen erfolgen diesbezüglich von einem Mitarbeiter der Entwicklungsabteilung der Firma *hsc solutions*. Eine Integration der über das Interface 1 angebotenen Funktionalitäten ist nicht vorgesehen.

Bereich Workflow-Instanzen

Workflow-Instanzen müssen innerhalb des ERP-Systems *hscERP* ausführbar sein. In diesem Kontext muss es eine Möglichkeit zur Anzeige aktiver sowie inaktiver Workflow-Instanzen geben. Die Übersicht dient vornehmlich zur Information der Systemadministratoren und ist im Allgemeinen nur für diese zugänglich. Die Übersicht soll im Wesentlichen einer einsehbaren Historie gleichen. Verschiedene Selektionskriterien sind hierbei zu unterstützen. Beispielsweise für eine gezielte Suche nach abgebrochenen bzw. von einem bestimmten Mitarbeiter ausgelösten Workflow-Instanzen. Parallel muss es möglich sein, Workflow-Instanzen über das *hscERP* zu beenden. Beispielsweise im Falle nicht mehr benötigter Workflow-Instanzen.

Bereich User-Task

Entsprechend dem vorherigen Bereich muss es eine Option zur Anzeige aller User-Tasks geben, welche ausschließlich Systemadministratoren vorbehalten ist. Systemad-

ministratoren sind in diesem Zusammenhang dazu berechtigt Statusänderungen vorzunehmen bzw. einem User-Task einen alternativen Mitarbeiter zuzuordnen. Ein Standardbenutzer ist ausschließlich dazu berechtigt die ihm zugewiesenen User-Tasks auszuführen.

Bereich Benutzerverwaltung

Für das ERP- und Workflow-Management-System soll eine gemeinsame Benutzerverwaltung existieren. Die Benutzerverwaltung soll diesbezüglich innerhalb des ERP-Systems *hscERP* erfolgen. Eine vollständige Übersicht aller Anwendungsfälle für die entsprechenden Bereiche befindet sich in den Anlagen unter Punkt C

3.3.3 Nichtfunktionale Anforderungen

Unter nichtfunktionale Anforderungen verbergen sich Qualitätsmerkmale zur Bewertung von Softwareprodukten, welche ausführlich in DIN 66272:1994-10 beschrieben werden.¹⁴¹ Die sechs Hauptmerkmale sind Funktionalität, Zuverlässigkeit, Benutzbarkeit, Effizienz, Änderbarkeit und Übertragbarkeit.

Funktionalität Alle geforderten Funktionen müssen nach der Integration ausführbar sein.

Zuverlässigkeit Nach der Integration muss die Workflow-Abarbeitung fehlerfrei funktionieren. Gleichzeitig gilt es sicherzustellen, dass die Integration des Workflow-Management-Moduls die Ausführung bereits bestehender Module nicht verhindert.

Benutzbarkeit Die Benutzbarkeit des Workflow-Management-Systems ist für den allgemeinen Benutzer transparent zu gestalten. Eine eventuelle Einarbeitungszeit ist durch die Verwendung bestehender Bedienkonzepte gering zu halten. Der Benutzer bzw. Systemadministrator ist auf einen Fehler in der Workflow-Abarbeitung hinzuweisen.

Effizienz Die Ausführung mehrerer Workflow-Instanzen darf die allgemeine Systemperformance nicht beeinflussen.

Änderbarkeit Es muss möglich sein zusätzliche Funktionen des Workflow-Management-Systems auf einfache Weise zu integrieren. Ebenso muss der Austausch des vollständigen Workflow-Management-Systems möglich sein.

Übertragbarkeit Das ERP-System *hscERP* muss nach der Integration des Workflow-Management-Systems weiterhin unter allen Betriebssystemen ausführbar sein.

¹⁴¹ Der Download ist kostenpflichtig.

4 Integrationskonzept für das hscERP

In den folgenden Abschnitten erfolgt eine Beschreibung des Integrationskonzepts für das Workflow-Management-System *jBPM* in Bezug auf das ERP-System *hscERP*. Ausgangspunkt der Überlegungen bilden die im vorangegangenen Kapitel gewonnenen Erkenntnisse aus der Analyse sowie die daraus abgeleiteten und definierten Anforderungen.

4.1 Architektur des Gesamtsystems

Das Gesamtsystem soll als Drei-Schichtenarchitektur realisiert werden, wobei ein paralleler Betrieb in einer Zwei-Schichtenarchitektur zu gewährleisten ist. Dies ist notwendig, da aktuelle Portierungsbemühungen andauern und der Zeitpunkt eines vollständigen Umstiegs auf eine Drei-Schichtenarchitektur nicht prognostiziert werden kann. Das Gesamtsystem besteht – in Anbetracht der Drei-Schichtenarchitektur – aus einer Client-, Server- sowie Datenbank-Komponente. Eine abstrahierte Übersicht der verschiedenen Komponenten sowie deren Bestandteile zeigt Abbildung 4.1.

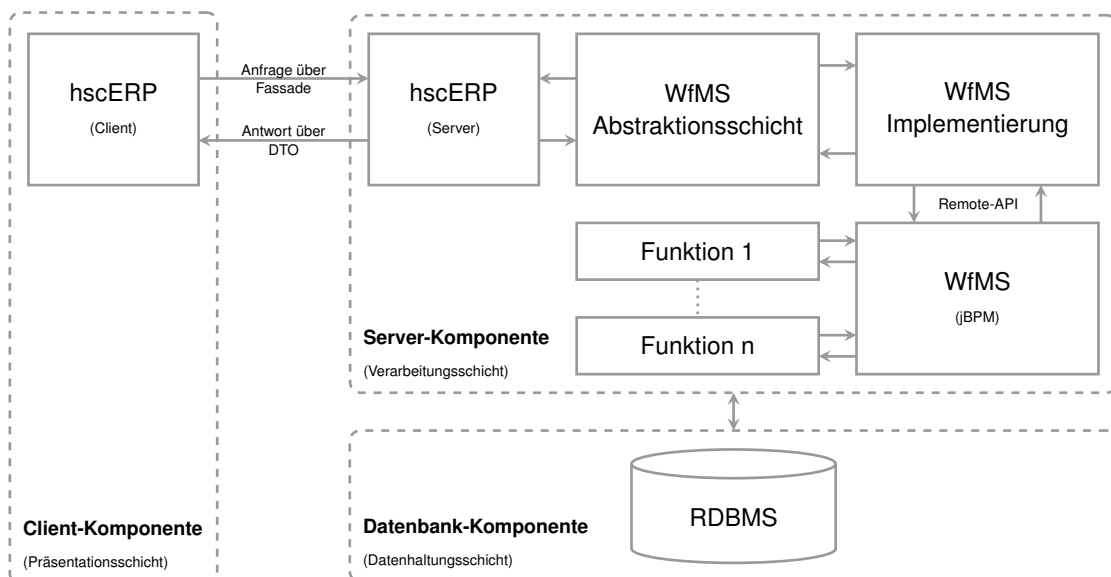


Abbildung 4.1: Abstrahierte Übersicht auf das Gesamtsystem

Jede einzelne Komponente repräsentiert eine Schicht in der Drei-Schichtenarchitektur. Aus Benutzersicht stellt die Client-Komponente diesbezüglich die erste Schicht dar. Eine konzeptionelle Beschreibung der Funktionalitäten sowie deren Zusammenspiel untereinander folgt in den kommenden Abschnitten.

4.2 Datenbank-Komponente

Die Datenbank-Komponente ist verantwortlich für die persistente Speicherung und Verwaltung der zur Ausführung von Workflow-Instanzen und User-Tasks benötigten Daten. Diese werden angrenzenden Schichten (z. B. der Datenhaltungsschicht) bzw. der Server-Komponente über entsprechende Schnittstellen (z. B. JDBC, ODBC) bereitgestellt. Das *hscERP* nutzt zur Speicherung und Verwaltung der Daten ein relationales Datenbank Management System. Um Datenobjekte des ERP-Systems *hscERP* in Form von Relationen in einer Datenbank speichern zu können, ist es möglich ein objektrelationales Mapping durchzuführen. Dabei werden die Objekte einer Klasse auf genau eine Datenbanktabelle abgebildet. Ein Objekt einer Klasse entspricht hierbei genau einer Zeile und die entsprechenden Attribute der Klasse bilden die Spalten. Innerhalb des ERP-Systems *hscERP* geschieht dies über einen eigens von der Firma *hsc solutions* entwickelten O/R-Mapper.

4.2.1 Datenmodell des Gesamtsystems

Das Workflow-Management-System *jBPM* verfügt in der Standardkonfiguration über eine separate Datenbank. Alternativ ist es möglich, alle von *jBPM* verwendeten Datenbanktabellen in die eigene Datenbank zu integrieren. Hierfür stellt *jBPM* verschiedene SQL-Skripte bereit, welche in Abhängigkeit des verwendeten RDBMSs (z. B. MySQL, PostgreSQL) die erforderlichen Datenbanktabellen anlegen. Eine Trennung der Datenbanken begünstigt hierbei den Aufwand einer späteren Lastverteilung. Beispielsweise kann es vorteilhaft sein eine logisch getrennte Datenbank auf einen physisch getrennten Datenbankserver auszulagern, um dadurch die Bearbeitungsgeschwindigkeit eingehender Anfragen zu erhöhen. Nach dem Austausch des Workflow-Management-Systems führen jedoch beide Fälle zu einem Problem.

Unter der Voraussetzung einer stetigen Analyse und Optimierung der Workflow-Modelle – in Bezug auf den Workflow-Regelkreis – können die Daten des vorherigen Workflow-Management-Systems weiterhin von Interesse sein. Beispielsweise für einen Vergleich der Bearbeitungsdauer von Workflow-Instanzen und User-Task zwischen den Systemen, aber auch darüber hinaus. Dabei ist davon auszugehen, dass sich die Bezeichnung und Anzahl der Datenbanktabellen des neuen Workflow-Management-Systems vom Vorgängersystem unterscheidet. Die bereitgestellte Spezifikation der Workflow Management Coalition beinhaltet lediglich Empfehlungen für den Aufbau eines Workflow-Management-Systems und ist für ein Herstellerunternehmen nicht bindend. Nach einem Austausch des Workflow-Management-Systems ist demnach in beiden Fällen die fortlaufende Unterstützung beider Systeme durch das *hscERP* erforderlich.

Um diesem Problem entgegenzuwirken sind Informationen, beispielsweise über die Bearbeitungsdauer von Workflow-Instanzen und User-Task, innerhalb des bestehenden

Datenbankmodelles des ERP-Systems *hscERP* zu integrieren. Die Datenintegration garantiert die Möglichkeit einer fortlaufenden Analyse der Workflow-Modelle ohne die Berücksichtigung des aktuell verwendeten Workflow-Management-Systems. Das diesbezüglich eingeführte Datenbankmodell ist in Abbildung 4.2 illustriert.

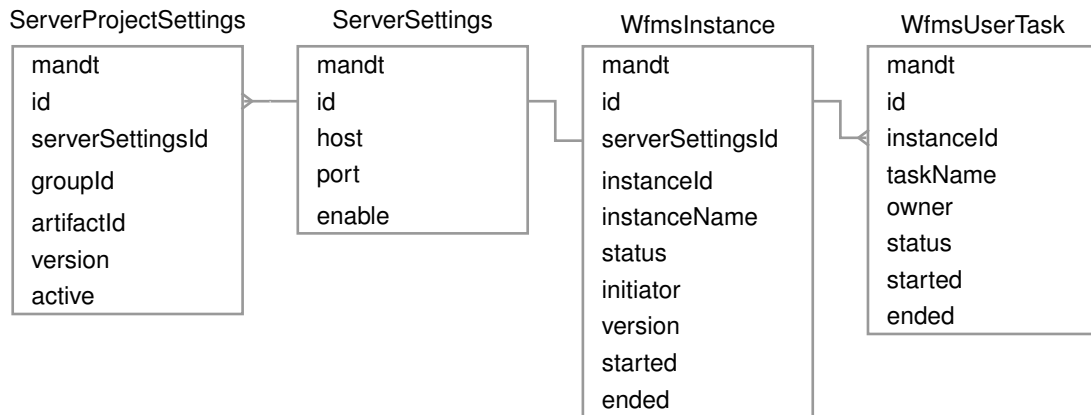


Abbildung 4.2: Datenbankmodell für die Integration des WfMS

Alle abgebildeten Tabellen werden später als eigenständige Klassen implementiert. Die Tabelle *ServerProjectSettings* enthält grundlegende Informationen über die auf dem Server existierenden Projekte. Alle Workflow-Modelle eines Anwenderunternehmens befinden sich hierbei in einem Server-Projekt, wobei ein Server-Projekt für verschiedenen Mandanten zur Verfügung gestellt werden kann. Die Attribute *groupId*, *artifactId* und *version* sind zwingend notwendig für einen Verbindungsaufbau zum Workflow-Management-System über die REST-API. Anhand der Version ist es möglich, Änderungen innerhalb des Server-Projektes zu veranschaulichen. Über das Attribut *active* sind Server-Projekte zu aktivieren bzw. deaktivieren. Workflow-Modelle innerhalb deaktivierter Server-Projekte stehen anschließend nicht mehr zur Ausführung bereit.

Die Tabelle *ServerSettings* enthält konkrete Angaben bezüglich *host* und *port* des Workflow-Management-Systems und vervollständigt somit die Verbindungsinformationen. Neben diesen Informationen ist, für einen erfolgreichen Verbindungsaufbau, eine Benutzerauthentifizierung erforderlich, welche Angaben über Name und Passwort verlangt. Diese Informationen werden über die bereits existierende Datenbanktabellen *User* und *Web_User* abgebildet. Da keine Änderungen an diesen Tabellen vorzunehmen sind, findet eine Darstellung im obigen Datenbankmodell nicht statt. Das Attribut *enable* erlaubt eine vollständige Deaktivierung des Workflow-Management-Systems.

Über die Tabelle *WfmsInstance* erfolgt eine Abbildung aller ausgeführten und aktiven Workflow-Instanzen. Jede Workflow-Instanz ist hierbei einem Eintrage in der Tabelle *ServerSetting* zuzuordnen. Das Attribut *instanceId* stellt eine Kombination aus *Group-Id*, *Artefakt-Id*, *Version* und dem Workflow-Modellnamen dar und dient im Wesentlichen zur Identifikation verschiedener Workflow-Instanzen des gleichen Typs. Das

Attribut `instanceName` entspricht dem Workflow-Modellnamen. Neben diesen Informationen bietet die Tabelle `WfmsInstance` weitere Attribute zur Identifikation des aktuellen Status einer Workflow-Instanz sowie den Namen des Initiators. Für die Auswertung der Bearbeitungsdauer enthält die Tabelle `WfmsInstance` zwei weitere Attribute, welche den Ausführungsbeginn und das Ausführungsende einer Workflow-Instanz kennzeichnen.

Eine Workflow-Instanz kann verschiedene Tasks enthalten, welche über die Tabelle `WfmsTask` abgebildet werden. Darin sind alle Tasks gespeichert, welche während der Workflow-Abarbeitung ausgeführt wurden bzw. aktuell ausgeführt werden. Dadurch ist es möglich den gegangenen Pfad innerhalb des Workflow-Modells im Rahmen einer Auswertung nachzuvollziehen. Zusätzlich ist es anhand der Statusinformation möglich laufende Workflow-Instanzen zu identifizieren. Diese Tatsache ist besonders in Zusammenhang mit der erforderlichen User-Task-Verwaltung von Interesse.

4.2.2 Datenintegritätsprüfung

Einer Workflow-Instanz bzw. einem User-Task können verschiedene Variablen übergeben werden. Mittels der übergebenen Variablen ist es beispielsweise möglich die interne Ablaufsteuerung einer Workflow-Instanz zu beeinflussen. Hierfür benötigte Variablen sind explizit im Workflow-Modell zu definieren. Variablen einer Workflow-Instanz können hierbei direkt über das angelegte Geschäftsobjekt (z. B. eine Rechnung) bezogen werden.

Die Eingabe der Daten erfolgt über Dialoge innerhalb des ERP-Systems *hscERP*. Um zu gewährleisten, dass ausschließlich korrekte Daten übergeben werden, ist eine Vollständigkeitsprüfung notwendig. Diese erfolgt innerhalb der Darstellungsschicht und somit in der Client-Komponente des *hscERP*. Eventuelle Pflichtfelder sind bereits vor der Vollständigkeitsprüfung im entsprechenden Eingabedialog als solche zu kennzeichnen. Fehlende Eingaben werden mit einer Fehlermeldung quittiert.

4.3 Server-Komponente

Die Server-Komponente befindet sich, in Bezug auf die Drei-Schichtenarchitektur, auf der Verarbeitungsschicht. Sie birgt die eigentliche Geschäftslogik für die Kommunikation mit dem Workflow-Management-System und somit zur Ausführung von Workflow-Instanzen und User-Tasks. Erforderliche Workflow-Instanz- bzw. User-Task-Variablen für die erfolgreiche Ausführung sind entsprechend nach bestandener Vollständigkeitsprüfung von der Präsentationsschicht an die Verarbeitungsschicht weiterzuleiten. Die Weiterleitung der Daten erfolgt hierbei über das Standardkonzept für den parallelen Betrieb einer Zwei- und Drei-Schichtenarchitektur mittels Fassaden-Objekt. Innerhalb der

Verarbeitungsschicht erfolgt anschließend die Übergabe der Daten an das Workflow-Management-System. Darin findet die Verarbeitung der Daten statt, wobei das Ergebnis unter anderem an die Präsentationsschicht zurück übermittelt wird. Gleichzeitig ist sie verantwortlich für die Transaktionssteuerung zwischen der darunterliegenden Datenhaltungsschicht. Beispielsweise für die persistente Speicherung der berechneten Ergebnisse.

Zur Umsetzung der eben beschriebenen Funktionalität verfügt die Server-Komponente über verschiedene Einzelkomponenten, welche im Folgenden genauer beschrieben werden.

4.3.1 hscERP-Komponente

Die hscERP-Komponente (Server) stellt die Verbindung zur Präsentations- und Datenhaltungsschicht her. Sie nimmt die Anfragen des Clients entgegen und führt entsprechende Datenbankoperationen aus. Hierzu zählen Operationen zum Speichern, Lesen, Ändern bzw. Löschen der Daten. In diesem Zusammenhang ist sie verantwortlich vor das benötigte O/R-Mapping. Gleichzeitig verwaltet sie die Verbindung zum Workflow-Management-System und ist zuständig für die Initialisierung und Manipulation von Workflow-Instanzen über die angrenzende *WfMS Abstraktionsschicht*.

4.3.2 WfMS Abstraktionsschicht

Die *WfMS Abstraktionsschicht* stellt eine Vielzahl abstrakter Schnittstellen bereit. Diese werden von der hscERP-Komponente zur Kommunikation mit dem angeschlossenen Workflow-Management-System verwendet. Im Fall von *jbPM* erfolgt die Kommunikation ausschließlich über die bereitgestellte Remote-API. Die *WfMS Abstraktionsschicht* dient dazu, die verwendete Remote-API vor der hscERP-Komponente zu verbergen. Im Ergebnis führt dies zu einer klaren Trennung zwischen Workflow-Management-System und *hscERP*. Bei einem Austausch des Workflow-Management-Systems bleibt somit die hscERP-Komponente unberührt. Die *WfMS Abstraktionsschicht* sorgt demnach für eine transparente Nutzung von Funktionalitäten des Workflow-Management-Systems sowie einer losen Kopplung. Gleichzeitig begünstigt sie die Konzentration auf zentral benötigte Funktionen, welche zwingend für die Kommunikation zwischen *hscERP* und Workflow-Management-System von Bedeutung sind. Dies minimiert den zukünftigen Entwicklungsaufwand, da lediglich erforderliche Funktionen bei einem Austausch neu zu implementieren sind. Die *WfMS Abstraktionsschicht* definiert diesbezüglich den Rahmen in Form einer abstrakten API. Dies kommt auch der Tatsache entgegen, wonach KUM im Allgemeinen über ein geringes finanzielles Budget für Investitionen verfügen und die Akzeptanz für eine spätere Einführung eines Workflow-Management-Systems bei Anwenderunternehmen durch geringe Anpassungskosten erhöht werden kann.

4.3.3 WfMS Implementierung

Bei der *WfMS Implementierung* handelt es sich um eine konkrete Implementierung der bereitgestellten Funktionalitäten eines Workflow-Management-Systems. In Bezug auf *jBPM* besitzt die *WfMS Implementierung* als alleinige Komponente Kenntnis über die verwendete Remote-API. Alle Funktionsaufrufe der Remote-API finden innerhalb der *WfMS Implementierung* statt. Jedes zu integrierende Workflow-Management-System verfügt über eine separate *WfMS Implementierung*. Bei einem Austausch des Workflow-Management-Systems ist lediglich die *WfMS Implementierung* auszutauschen.

4.4 Client-Komponente

Die Client-Komponente dient als Benutzerschnittstelle und somit als zentraler Zugangspunkt für einen Benutzer. Sie ermöglicht die Ausführung und Verwaltung von User-Tasks sowie Workflow-Instanzen. In Abhängigkeit der zugehörigen Rolle eines Benutzers, ist diesem lediglich Zugriff auf im Vorfeld definierte Funktionen des Workflow-Management-Systems zu gewähren. Diese sind dem Benutzer in der hscERP-Komponente (Client) über zwei unterschiedliche Bereiche zur Verfügung zu stellen.

Systemadministratoren (Admin) soll der alleinige Zugriff auf die Verwaltungsfunktionen des Workflow-Management-Systems gestattet sein. Alle Verwaltungsaspekte sind diesbezüglich über ein zusätzliches Workflow-Management-Modul abzudecken.

Standardbenutzer (Benutzer) dürfen hingegen lediglich die ihnen zugeteilten User-Tasks ausführen und Workflow-Instanzen initialisieren, wobei eine Initialisierung vollkommen transparent für den Benutzer geschehen soll. Die benutzerspezifische Anzeige und Ausführung der User-Tasks erfolgt dabei über eine separat erreichbare Worklist. Diese enthält alle ausführbaren User-Tasks für einen Benutzer. Abbildung 4.3 veranschaulicht den eben beschriebenen Sachverhalt noch einmal grafisch.

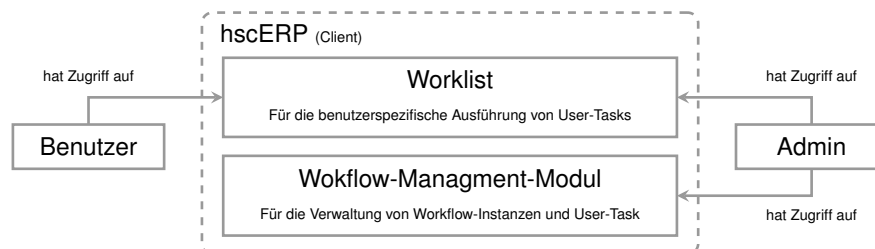


Abbildung 4.3: Zugangsbereiche für unterschiedliche Benutzerrollen

Die Realisierung der Zugriffssteuerung einzelner Bereiche geschieht in diesen Zusammenhang über die bestehende Benutzer- und Rechteverwaltung.

4.4.1 Worklist

Im Ergebnis dürfen Benutzer ausschließlich Zugriff auf die Worklist und den ihnen zugewiesenen User-Tasks erhalten. Systemadministratoren sind im Gegensatz dazu berechtigt die User-Tasks aller Benutzer einzusehen. Um eine Übersicht aller aktiven User-Tasks zu erhalten, sendet die Client-Komponente über ein Fassaden-Objekt eine Anfrage inklusive Benutzerinformationen an die Server-Komponente. In dieser werden innerhalb der hscERP-Komponente (Server) die entsprechenden Methoden für die *WfMS Abstraktionsschicht* aufgerufen, welche die Anfrage an eine konkrete *WfMS Implementierung* weiterleitet. Die *WfMS Implementierung* erstellt aus den übergebenen Informationen eine präzise Anfrage für das aktuell aktive Workflow-Management-System und leitet diese entsprechend weiter. Nach Bearbeitung der Anfrage durch das Workflow-Management-System übermittelt dieses eine Antwort mit einer Liste aller aktiven User-Tasks an die *WfMS Implementierung*. Diese übernimmt ein individuelles Mapping der erhaltenen Liste auf die benötigten DTO-Objekte. Nach erfolgreichem Mapping wird das Ergebnis der Client-Komponente bereitgestellt.

Der Funktionsumfang der Worklist beschränkt sich hierbei auf das Anzeigen und Verwalten aller aktiven User-Tasks. Von besonderem Interesse sind in diesem Zusammenhang User-Tasks in den Status *Created*, *Ready*, *Reserved* und *InProgress*. Die Statusvergabe findet hierbei automatisch durch das Workflow-Management-System statt. Ein User-Task im Status *Created* spiegelt hierbei einen neu angelegten User-Task wider. User-Tasks im Status *Created* sind lediglich für Systemadministratoren sichtbar und von ebendiesen explizit einem Benutzer zuzuweisen. Ein User-Task im Status *Ready* signalisiert einen User-Task einer Benutzergruppe. Jeder Benutzer dieser Gruppe ist berechtigt diesen User-Task auszuführen. Ein User-Task im Status *Reserved* wurde von einem Benutzer reserviert und steht daraufhin der Gruppe nicht mehr zur Verfügung. Ein reservierter User-Task kann in diesem Zusammenhang zu jeder Zeit durch den Benutzer zurück in den Status *Ready* versetzt werden, wonach er allen Benutzern der Gruppe erneut zu Verfügung steht. Nachdem sich ein Benutzer einen User-Task reserviert hat, ist er berechtigt diesen auszuführen. Daraufhin wechselt der User-Task in den Status *InProgress* und es öffnet sich ein Dialog zur Eingabe eventuell benötigter Daten. Nach Eingabe aller notwendigen Daten erfolgt eine Vollständigkeitsprüfung durch die Client-Komponente. Fehlende Eingaben werden mit einem Hinweis quittiert und der Benutzer erhält die Möglichkeit zur Vervollständigung der fehlenden Daten. Nach vollständiger Eingabe aller benötigten Daten erfolgt eine Übergabe dieser an den User-Task, woraufhin der User-Task nicht mehr in der Worklist erscheint. Gleichzeitig erfolgt eine Speicherung des abgearbeiteten User-Tasks in der relationalen Datenbank des *hscERP*, zur späteren Auswertung.

4.4.2 Workflow-Management-Modul

Die Einbindung des Workflow-Management-Moduls erfolgt über einen separaten Menüeintrag innerhalb der hscERP-Komponente. Die Modulfunktionalitäten sind ausschließlich für Systemadministratoren zugänglich. Über ein Untermenü sind hierbei verschiedene Funktionalitäten des Workflow-Management-Systems zu erreichen. Hierzu zählen beispielsweise Funktionen zur Verwaltung von Servereinstellungen in Bezug auf das zu nutzende Workflow-Management-System. Weiterhin sind hierunter die verschiedenen Server-Projekte, welche die Workflow-Modelle beinhalten, konfigurierbar. Neben einer einfachen Systemkonfiguration können Systemadministratoren in diesem Bereich alle aktiven und bisher fertiggestellten Workflow-Instanzen sowie User-Tasks eingesehen und ausgewertet werden.

5 Prototypische Implementierung

Im Rahmen einer prototypischen Implementierung wurde das im letzten Kapitel beschriebene Integrationskonzept vollständig umgesetzt. Dieses Kapitel wird diesbezüglich wesentliche Aspekte der Implementierung vorstellen.

5.1 Realisierung der Datenhaltung

Das im Integrationskonzept beschriebene Datenmodell zur persistenten Speicherung der Servereinstellungen sowie Workflow-Instanz- und User-Task-Informationen besteht aus vier Tabellen. Für alle vier Tabellen wurden entsprechende DTO-Objekte angelegt, welche über das O/R-Mapping mit Daten aus der Datenbank gefüllt werden.

5.2 Realisierung der Server-Komponente

Die Hauptarbeit bei der Realisierung der Server-Komponente bestand in der Bereitstellung einer Abstraktionsschicht für Workflow-Management-Systeme. Die wesentlichen Aufgaben der *WfMS Abstraktionsschicht* bestehen dabei in der transparenten Unterstützung des Verbindungsaufbaues sowie der Ausführung von Workflow-Instanzen und User-Tasks.

5.2.1 Implementierung des Verbindungsaufbaus

Die abstrakte Klasse `WfmsConnector` definiert Methoden für die Verwaltung der Verbindung zum Workflow-Management-System und wird innerhalb eines Bean-Objektes verwendet. Die vier darüber bereitgestellten Methoden lauten:

- `connect()`
- `disconnect()`
- `isConnected()`
- `getWfmsEngine()`

Die Methoden `connect()` und `disconnect()` sind verantwortlich für den Aufbau bzw. Abbau der Verbindung zum Workflow-Management-System. Der aktuelle Verbindungsstatus ist über die Methode `isConnected()` prüfbar. Für jedes Workflow-Management-System ist in diesem Zusammenhang eine entsprechende Connector-Klasse abzuleiten und die Methoden `connect()`, `disconnect()` sowie `getWfmsEngine()` zu imple-

mentieren. Eine korrekte Implementierung besitzt Abhängigkeiten zu einem User- sowie ServerSettings-Objekt. Ein User-Objekt ist notwendig, um den aktuell am ERP-System angemeldeten Benutzer zu identifizieren. Das ServerSettings-Objekt enthält die genauen Verbindungsinformationen zum Workflow-Management-System und den darüber ausführbaren Workflow-Instanzen. Abbildung 5.1 stellt das Beschriebenen noch einmal modellhaft dar.

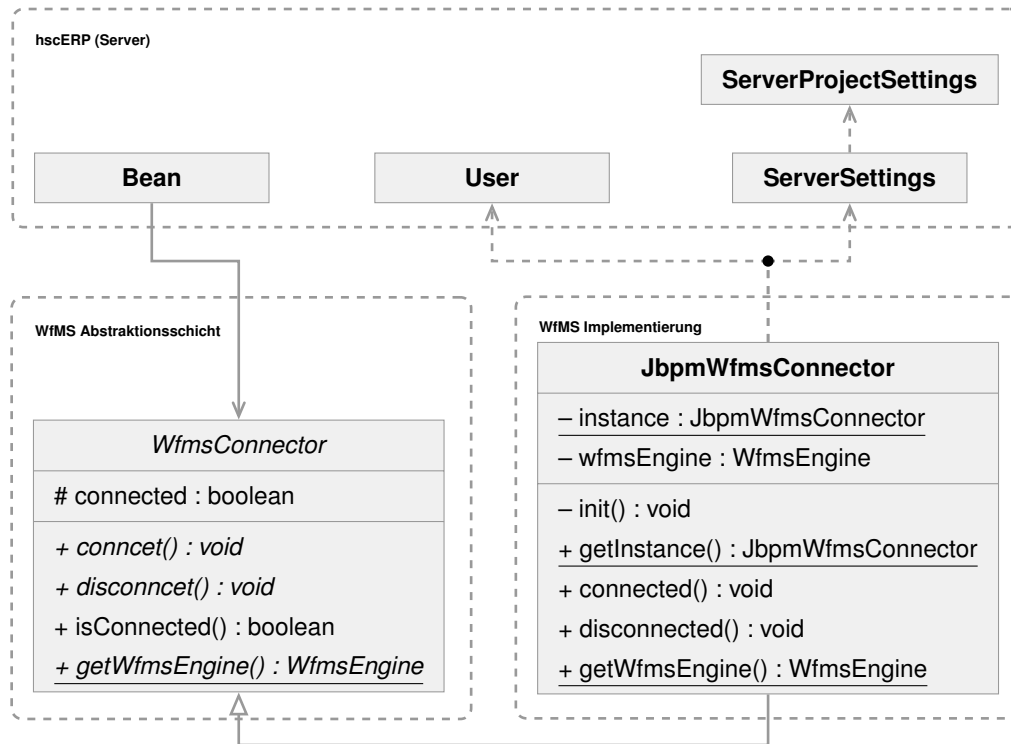


Abbildung 5.1: Implementierung des grundlegenden Verbindungsaufbaus

Die Klasse *JbpmWfmsConnector* ist Bestandteil der *WfMS Implementierung* und stellt eine konkrete Implementierung der *WfmsConnector*-Klasse dar. Die *init()*-Methode initialisiert zunächst alle benötigten Verbindungsinformationen. User-Informationen werden diesbezüglich aus der aktuellen Session des *hscERPs* geladen. Informationen über die aktuell aktiven Servereinstellungen sind über die Datenbank zu beziehen und über die beiden Objekte *ServerSettings* und *ServerProjectSettings* bereitzustellen. In der Methode *connect()* wird die *RuntimeEngine* von *jBPM* anschließend initialisiert. Nach der Initialisierung kann über die Methode *getWfmsEngine* ein neues *WfmsEngine*-Objekt erzeugt werden. Bei der Initialisierung des *WfmsEngine*-Objekts wird diesem die *RuntimeEngine* übergeben.

5.2.2 Implementierung des Workflow-Instanz- und User-Task-Zugriffs

Bei dem `WfmsEngine`-Objekt handelt es sich um ein Interface der *WfMS Abstraktionsschicht*. Für die Verwaltung von Workflow-Instanzen und User-Tasks definiert das Interface zwei zentrale Methoden mit der Bezeichnung

- `getInstanceManager()` und
- `getUserTaskManager()`.

Eine Implementierung erfolgt innerhalb der Klasse `JbpmWfmsEngine`, wie in Abbildung 5.2 grafisch veranschaulicht.

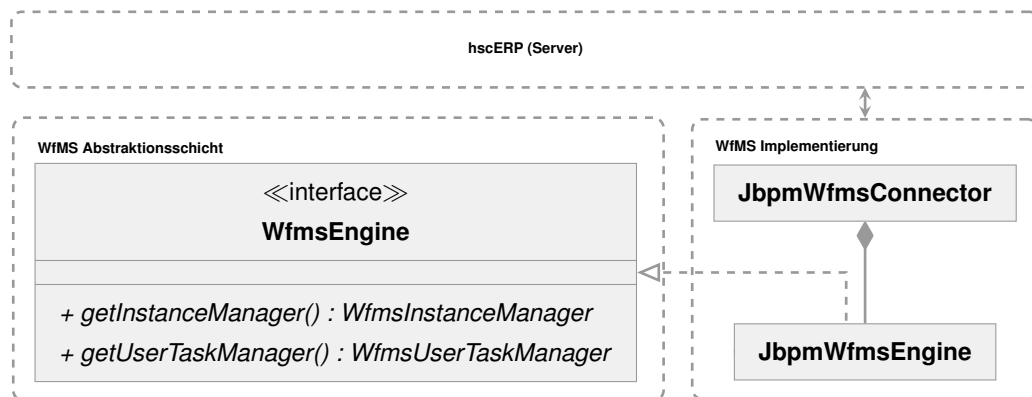


Abbildung 5.2: Implementierung der Workflow-Instanz- und User-Task-Verwaltung

Die Beziehung zwischen `JbpmWfmsConnector` und `JbpmWfmsEngine` ist als unidirektionale Komposition dargestellt. Ein Objekt der Klasse `JbpmWfmsEngine` kann demnach nicht ohne ein vorher erzeugtes `JbpmWfmsConnector`-Objekt existieren. Gleichzeitig ist die Existenz eines `JbpmWfmsEngine`-Objekts keine zwingende Voraussetzung für die Existenz eines `JbpmWfmsConnector`-Objekts. Innerhalb des `JbpmWfmsEngine`-Objekts werden die entsprechenden Manager-Objekte initialisiert. Bei der Initialisierung des `WfmsInstanceManager`- bzw. `WfmsUserTaskManager`-Objekts wird in diesem Zusammenhang die `KieSession` bzw. der `TaskService` von *jBPM* übergeben. Dieses Vorgehen sorgt dafür, dass bei einem späteren Austausch des Workflow-Management-Systems lediglich der Connector auszutauschen ist. Der in Listing 5.1 gezeigte Quellcodeausschnitt soll dies verdeutlichen.

Listing 5.1: Beispiel für die Ausführung einer Workflow-Instanz über die abstrakte API

```

1 WfmsConnector connector = JbpmWfmsConnector.getInstance();
2 connector.connect();
3 WfmsInstanceManager instanceManager = connector.getWfmsEngine().
  getWfmsInstanceManager();
4 instanceManager.startWorkflowInstance("example.processId")

```


Wechselt man in der ersten Zeile den `JbpmWfmsConnector` durch eine beliebige alternative Implementation der `WfmsConnector`-Klasse aus, werden ab sofort alle Workflow-Instanzen über das alternative Workflow-Management-System ausgeführt. Das gleiche Prinzip findet in Bezug auf das `WfmsUserTaskManager`-Interface Anwendung. Die über die Interfaces `WfmsInstanceManager` und `WfmsUserTaskManager` bereitgestellten Funktionalitäten können dabei auf einfache Weise erweitert werden. Diesbezüglich sind entsprechende Methoden in den Interface-Klassen zu definieren und anschließend zu implementieren. Den aktuellen Funktionsumfang des `WfmsUserTaskManager`-Interfaces zeigt Abbildung 5.3.

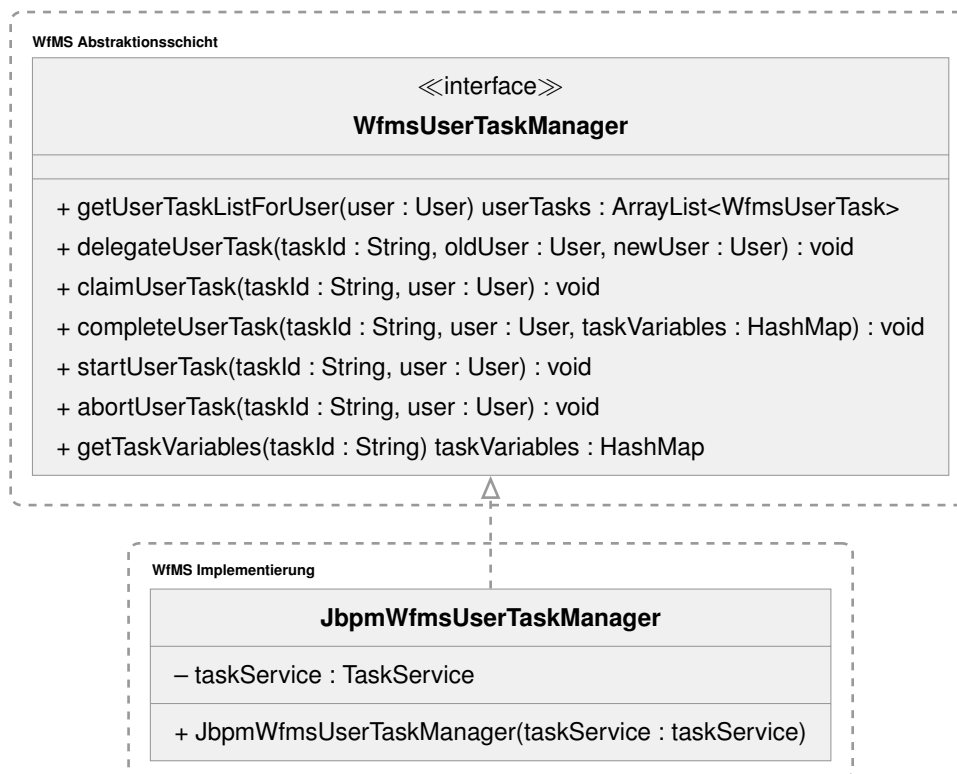


Abbildung 5.3: Implementierung User-Task-Manager

Die Methode `getUserTaskListForUser(user : User)` liefert eine Liste aller User-Tasks des aktuell am ERP-System angemeldeten Benutzers. Für die Selektion der relevanten User-Task stellt die Remote-API über das Interface `TaskService` die Methode

- `getTasksAssignedAsPotentialOwner(String userId, String language)`

bereit. Diese besitzt als Rückgabewert eine Liste von `TaskSummary`-Objekten. In der Methode `getUserTaskListForUser(user : User)` erfolgt diesbezüglich ein Mapping aller Objekte auf `WfmsUserTask`-Objekte. Das Ergebnis wird als `ArrayList` abgelegt und kann anschließend von der Client-Komponente zur weiteren Verarbeitung verwendet werden.

Eine weitere zentrale Methode des Interfaces `WfmsUserTaskManager` stellt die Hilfsme-

thode `getTaskVariables(taskId : String)` dar. Sie liest die aktuellen User-Task-Variablen eines User-Tasks aus. Die Remote-API stellt diesbezüglich über das Interface `TaskService` die Methode

- `getTaskContent(taskId : String)`

zur Verfügung. Sie liefert eine Sammlung von Schlüssel-Wert-Paaren, welche der Client-Komponente zur Verfügung zu stellen sind und zur die Anzeige innerhalb eines User-Task-Dialogs Verwendung finden. Die weiteren in Abbildung 5.3 gezeigten Methoden dienen der direkten Manipulation der User-Tasks und haben stets eine Satzungsänderung zur Folge. In diesem Kontext ist die Methode `completeUserTask(taskId : String, taskVariables : HashMap)` von zentraler Bedeutung. Sie gestattet die Manipulation von User-Task-Variablen durch den Benutzer während der Ausführung.

Wie bereits oben angedeutet dient das `WfmsEngine`-Interface als Schnittstelle zur Verwaltung der User-Tasks und Workflow-Instanzen. Den aktuellen Funktionsumfang in Bezug auf Workflow-Instanzen zeigt Abbildung 5.4.

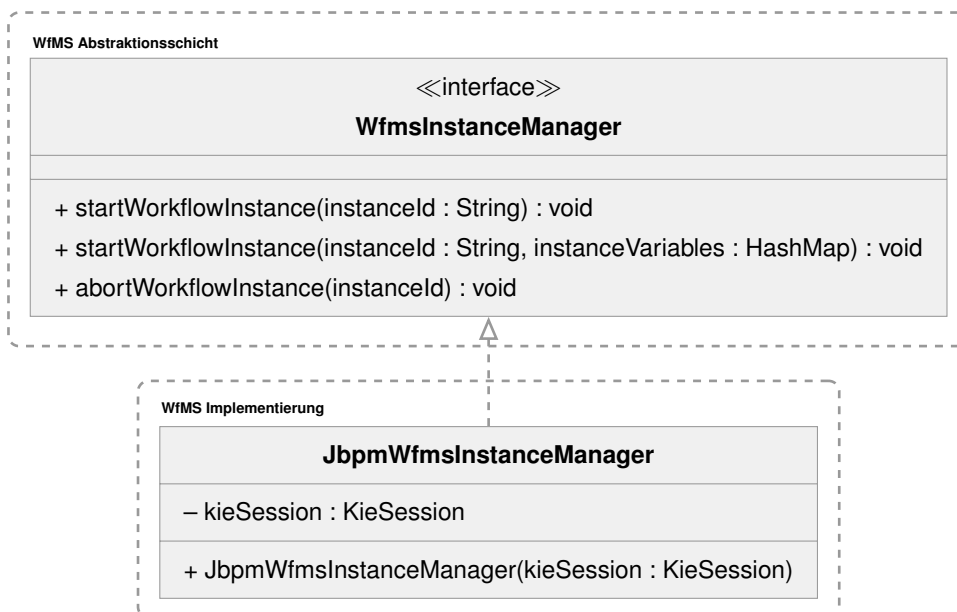


Abbildung 5.4: Implementierung Instance-Manager

Das Interface ist im Wesentlichen für den Start und Abbruch von Workflow-Instanzen verantwortlich. Diesbezüglich wird eine gültige `instanceId` benötigt. Diese entspricht im Wesentlichen dem Namen des Workflow-Modells. Vor der Übergabe der `instanceId` wird zunächst innerhalb der Bean geprüft, ob die übergebene `instanceId` existiert. Hierfür wird das `ServerSettings`-Objekt nach passender Übereinstimmung durchsucht. Bei einer Übereinstimmung wird im Anschluss die entsprechende Start-Methode des `WfmsInstanceManager`-Interfaces aufgerufen.

5.3 Realisierung der Client-Komponente

Die Client-Komponente bietet die Möglichkeit zu Manipulation von Workflow-Instanzen und User-Tasks. Für die Realisierung verwendet sie die bereitgestellten Funktionalitäten der beiden Schnittstellen `WfmsInstanceManager` und `WfmsUserTaskManager`. Für jeden User-Task ist in diesem Zusammenhang eine `WfmsUserTaskDialog`-Klasse zu implementieren. Jede Dialog-Klasse dient zur Anzeige und Manipulation der User-Task-Variablen eines bestimmten User-Tasks. Jedes Eingabefeld entspricht dabei einer Variable. Zur Verarbeitung der Eingaben bzw. Ausgaben verfügt sie über eine eigene Controller-Klasse, welche die Interface-Funktionalitäten implementiert und sie an ein Fassaden-Objekt zur weiteren Verarbeitung delegiert. In Abhängigkeit der unterstützten Architektur entscheidet das Fassaden-Objekt über die weitere Ausführung innerhalb eines Bean- bzw. FakeBean-Objekts.

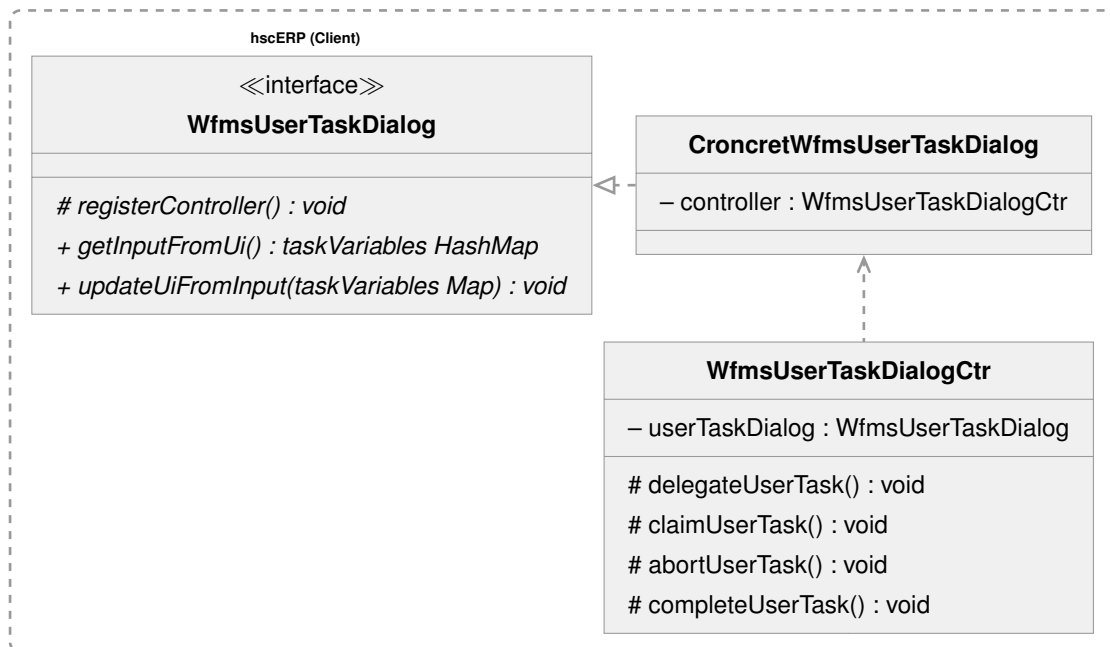


Abbildung 5.5: Implementierung des Client-Zugriffs

Das Interface `WfmsUserTaskDialog` definiert die wesentlichen Methoden im Umgang mit User-Task-Dialogen. Hierzu gehört eine Methode zur Registrierung der verwendeten Controller-Klasse sowie zwei Methoden zur Ein- bzw. Ausgabe der User-Task-Variablen. Die Methoden sind in der entsprechenden Unterklasse eines User-Task-Dialogs zu implementieren. Für jeden User-Task ist diesbezüglich eine separate Dialog-Klasse anzulegen. Die für einen speziellen User-Task benötigten Eingabefelder werden in der entsprechenden Unterklasse definiert. Die Klasse `WfmsUserTaskDialogCtr` definiert die Methoden zum Zugriff auf die `WfmsEngine`. Die aufgerufene Funktionalität wird diesbezüglich zunächst über die Fassade zur weiteren Verarbeitung an ein FakeBean- bzw. Bean-Objekt delegiert. Dies garantiert, dass die Client-Komponente zu keiner Zeit die Remote-API von *jBPM* verwendet. Bei einem Austausch des Workflow-Management-Systems ist demnach keine Änderung an der Client-Komponente vorzunehmen.

6 Zusammenfassung und Ausblick

Ziel der Arbeit war es, über das ERP-System *hscERP* ausgewählte Funktionalitäten eines Workflow-Management-Systems bereitzustellen. Durch die Einführung einer Abstraktionsschicht wurde dabei eine abstrakte Entwickler-API geschaffen, welche die Ausführung von Workflow-Instanzen und User-Tasks gestatten.

Die abstrakte API wurde so konzipiert, dass ein einfacher Austausch des Workflow-Management-Systems zu jeder Zeit möglich ist. Diesbezüglich fördert die abstrakte API die lose Kopplung zwischen Workflow-Management-System und ERP-System. Innerhalb des *hscERPs* werden zu keiner Zeit direkte Funktionen der angebotenen Remote-API von *JBPM* verwendet. Alle Funktionsaufrufe erfolgen über die eingeführte Abstraktionsschicht. Gleichzeitig definiert die API alle wesentlichen Funktionen, welche von einem neu zu integrierenden Workflow-Management-System zu unterstützen sind. Hierdurch wird ein klarer Rahmen für einen zukünftigen Austausch des Workflow-Management-Systems vorgegeben.

Ein aktuelles Problem betrifft vor allem die Auswertung der abgearbeiteten Workflow-Instanzen und User-Task. Aktuell wurden lediglich Verwaltungsfunktionen implementiert. Ausführliche Reporting-Funktionen sind jedoch besonders in Bezug auf die Analyse und Optimierung von Workflow-Modellen hilfreich. An verschiedenen Stellen im *hscERP* wird zur Anfertigung von Reporten die Bibliothek JasperReports verwendet. Dabei handelt es sich um eine Open-Source Java-Library, welche unter anderem grafisch anschauliche Reporte als PDF-Datei erzeugen kann. Die Nutzung von JasperReports zur grafischen Auswertung der Workflow-Instanz- und User-Task-Abarbeitung erscheint daher als sinnvolle Erweiterung.

Anhang A: WfMC Metamodell

Die nachfolgende Auflistung wurde vollständig aus [Coa95, vgl. S. 30 f.] übernommen und ins Deutsche übersetzt. Sie beinhaltet eine Aufzählung relevanter Attribute des vorgestellten Metamodells aus Kapitel 2 Grundlagen.

Workflow Type Definition

- Workflow-Prozessname
- Versionsnummer
- Prozess Start- und Abschlussbedingung
- Sicherheits-, Audit oder sonstige Kontrolldaten

Activity

- Aktivitätsname
- Aktivitätstyp (User-Task, Script-Task, Sub-Workflow, etc.)
- Vor- und Nachgelagerte Aktivitätsbedingungen
- Weiter Ablaufeinschränkungen

Transition Conditions

- Ablauf- oder Ausführungsbedingung

Workflow relevant data

- Name der Daten und Dateipfad
- Datentyp

Role

- Name und Organisationseinheit

Invoked Application

- Generischer Typ oder Name
- Ausführungsparameter
- Speicherort bzw. Zugangsweg

Anhang B: WAPI-Funktionen Invoked Applications

Session Establishment

- Verbindung / Trennung von Sessions zwischen beteiligten Systemen

Workflow Definition Operations

- Abfragefunktionen (mit optionalen Auswahlkriterien) nach Workflow-Modellnamen oder Attributen

Process Control Functions

- Erzeugung / Starten / Beenden einer individuellen Workflow-Instanz
- Suspension / Wiederaufnahme einer individuellen Workflow-Instanz
- Erzwingen einer Statusänderung innerhalb einer individuellen Workflow-Instanz bzw. Aktivitätsinstanz (z. B. User-Task)
- Zuweisung bzw. Abfrage eines Attributes (z. B. Priorität) einer Workflow- bzw. Aktivitätsinstanz

Process Status Functions

- Öffnen / Schließen einer Workflow-Instanzen bzw. Aktivitätsinstanzabfrage mit optional einstellbaren Filterkriterien
- Detailabfragen zu Workflow-Instanzen bzw. Aktivitätsinstanzen nach spezifizierten Filterkriterien
- Detailabfragen zu einer spezifischen (individuellen) Workflow-Instanzen bzw. Aktivitätsinstanz

Worklist/Workitem Handling Functions

- Öffnen / Schließen einer Worklist-Abfrage mit optional einstellbaren Filterkriterien
- Abfrage von Workitems mit spezifizierten Filterkriterien
- Mitteilung über Auswahl / Zuordnung / Abschluss eines (spezifischen) Workitems
- Zuweisung bzw. Abfrage von Workitem-Attributen

Process Supervisory Functions

- Ändern des Betriebsstatus eines Workflow-Modells und/oder seiner noch vorhandenen Workflow-Instanzen
- Änderung des Status aller Workflow-Instanzen oder Aktivitätsinstanzen eines bestimmten Typs
- Zuordnung von Attributen zu allen Workflow-Instanzen oder Aktivitätsinstanzen eines bestimmten Typs
- Beendigung aller Workflow-Instanzen

Data Handling Functions

- Empfang / Rückgabe Workflow relevanter Daten

Administration Functions

- Unterstützung zusätzlicher Verwaltungsfunktionen über das WAPI ist möglicherweise geeignet für entsprechende Client-Anwendungen

Application Invocation

- Relevante Kommandos (z. B. Zugriff auf Process-, Aufgaben-, Workitem-Attribute und allgemeinen relevante Daten für einen Workflow)

Anhang C: Überblick aller Anwendungsfälle

Abbildung C.1 zeigt eine Übersicht aller Anwendungsfälle für das Workflow-Management-Modul. Dieses ist über einen gesonderten Menüeintrag im linken Bereich des *hscERP* erreichbar.

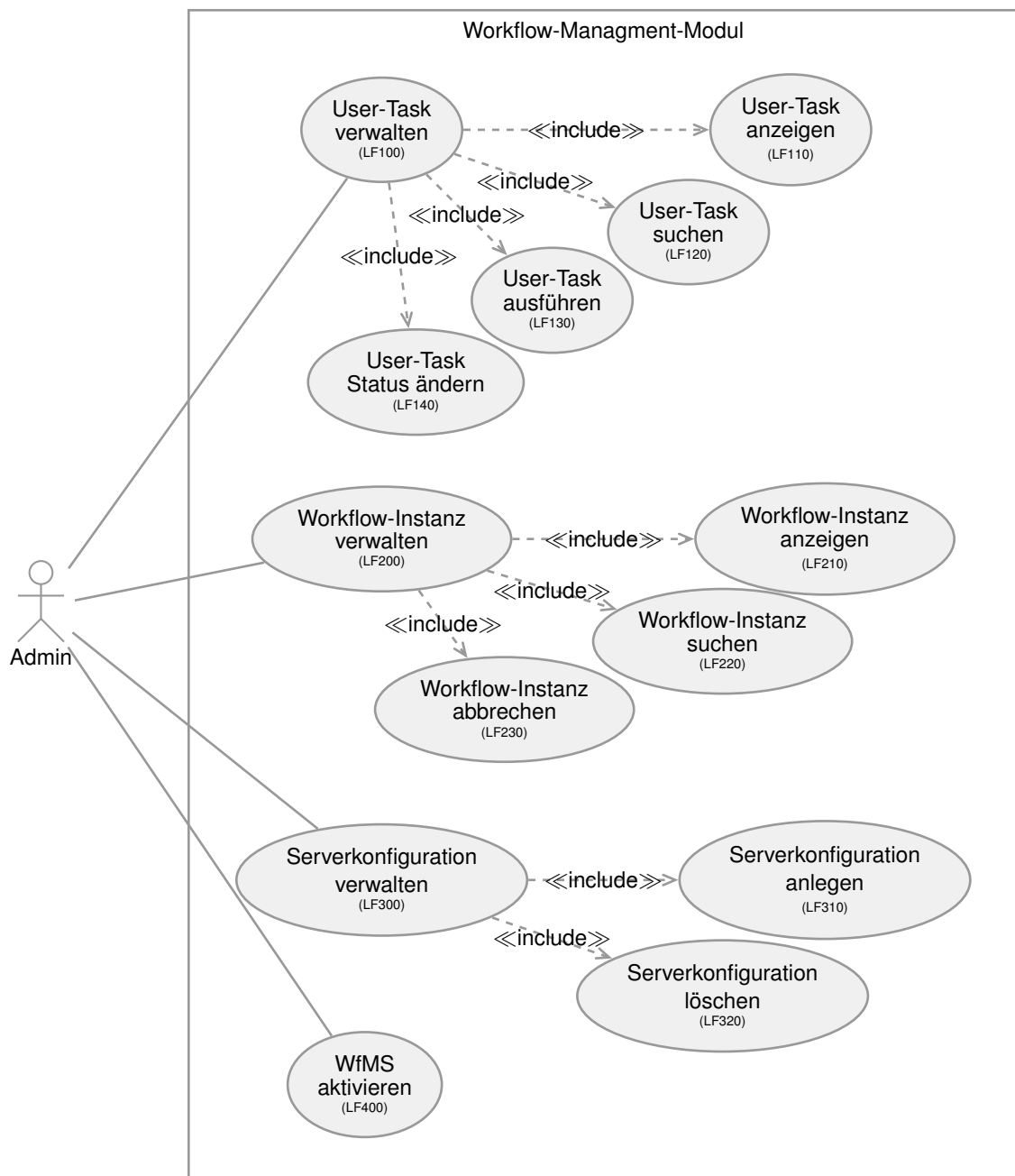


Abbildung C.1: Übersicht der Anwendungsfälle für das Workflow-Management-Modul

Abbildung C.2 zeigt eine Übersicht der Anwendungsfälle für den Bereich der Worklist, welche innerhalb des ERP-Systems *hscERP* über das Hauptmenü aufrufbar ist.

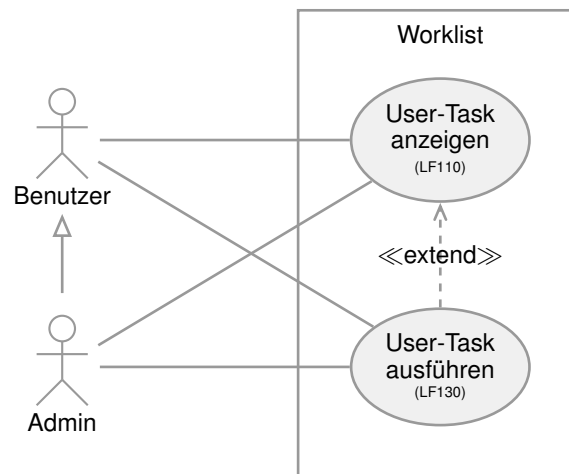


Abbildung C.2: Übersicht der Anwendungsfälle für die Worklist

Literaturverzeichnis

- [Coa94] Coalition, Workflow Managment: *Workflow Managment Coalition – Terminology & Glossary*, 1994. http://www.wfmc.org/standards/docs/TC-1011_term_glossary_v3.pdf.
- [Coa95] Coalition, Workflow Management: *Workflow Management Coalition – The Workflow Reference Model*, 1995. <http://www.wfmc.org/standards/docs/tc003v11.pdf>.
- [Dud16] Duden: *Duden der deutschen Rechtschreibung — Online Ausgabe*, 2016. <http://www.duden.de/>.
- [FG15] Fiorini, Simone und Arun V Gopalakrishnan: *Mastering jBPM6 - Design, build, and deploy business process-centric applications using the cutting-edge jBPM technology stack*. Packt Publishing Ltd, Birmingham, 2015, ISBN 978-1-78328-957-8.
- [fMB16] Mittelstandforschung Bonn, Institut für: *Die 10 häufigsten Fragen rund um den deutschen Mittelstand*, 2016. <http://www.ifm-bonn.org/definitionen/faq-fragen-zum-mittelstand>.
- [Fou04] Foundation, Apache Software: *Apache License Version 2.0*, 2004. <http://www.apache.org/licenses/LICENSE-2.0>.
- [Gad12] Gadatsch, Andreas: *Grundkurs Geschäftsprozess-Management - Methoden und Werkzeuge für die IT-Praxis: Eine Einführung für Studenten und Praktiker*. Springer-Verlag, Berlin Heidelberg New York, 7. Aufl. Auflage, 2012, ISBN 978-3-834-82428-8.
- [Gmb11] GmbH, Konradin Business: *Einsatz von ERP-Lösungen in der Industrie*, 2011. http://www.mav-online.de/c/document_library/get_file?uuid=8cd31198-8563-4db8-8414-3098e2e4408e&groupId=32571331.
- [Gmb15] GmbH, Statista: *Anzahl der Unternehmen in Deutschland nach Beschäftigtengrößenklassen im Jahr 2013 (Stand Mai 2015)*, 2015. <http://de.statista.com/statistik/daten/studie/1929/umfrage/unternehmen-nach-beschaefigtengroessenklassen/>.
- [Hag04] Hagen, Cornelia Richter von: *Business-Process- und Workflow-Management - Prozessverbesserung durch Prozess-Management*. Vieweg+Teubner Ver-

lag, Wiesbaden, 2004, ISBN 978-3-519-00491-2.

- [Hel15] Hellingrath, Prof. Dr. Ing. Bernd: *Supply Chain Management-System — Enzyklopädie der Wirtschaftsinformatik*, 2015. <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/informationssysteme/crm-scm-und-electronic-business/Supply-Chain-Management/Supply-Chain-Management-System>.
- [IS14] Immerschitt, Wolfgang und Marcus Stumpf: *Employer Branding für KMU - Der Mittelstand als attraktiver Arbeitgeber*. Springer-Verlag, Berlin Heidelberg New York, 1. Aufl. Auflage, 2014, ISBN 978-3-658-01204-5.
- [Kom03] Kommission, Europäische: *Definition der Kleinstunternehmen sowie der kleinen und mittleren Unternehmen*, 2003. <http://www.foerderdatenbank.de/Foerder-DB/Navigation/Foerderrecherche/suche.html?get=4aa561e46fff16fb87d819d09c769842;views;document&doc=2018&typ=KU>.
- [Lei12] Leiting, Andreas: *Unternehmensziel ERP-Einführung - IT muss Nutzen stiften*. Springer-Verlag, Berlin Heidelberg New York, 2012, ISBN 978-3-834-94462-7.
- [Lei15a] Leimeister, Jan Marco: *Einführung in die Wirtschaftsinformatik*. Springer-Verlag, Berlin Heidelberg New York, 12. Aufl. Auflage, 2015, ISBN 978-3-540-77847-9.
- [Lei15b] Lein, René: *Analyse von Architekturansätzen zur Performanceoptimierung komplexer Softwareanwendungen im KMU-Bereich*. Masterarbeit, Westsächsische Hochschule Zwickau, 2015.
- [Mer12] Mertens, Peter: *Integrierte Informationsverarbeitung 1 - Operative Systeme in der Industrie*. Springer-Verlag, Berlin Heidelberg New York, 18. Aufl. Auflage, 2012, ISBN 978-3-834-94395-8.
- [Mey15] Meyer, Prof. Dr. Silke: *Operative Informationssysteme*. Vorlesung, Hochschule Mittweida - University of Applied Sciences, 2015.
- [MSA14] Maio, Mariano Nicolas De, Mauricio Salatino und Esteban Aliverti: *jBPM6 Developer Guide - Learn about the components, tooling, and integration points that are part of the JBoss Business Process Management (BPM) Framework*. Packt Publishing Ltd, Birmingham, 2014, ISBN 978-1-78328-661-4.

- [Mü06] Müller, Joachim: *Workflow-based Integration - Grundlagen, Technologien, Management*. Springer-Verlag, Berlin Heidelberg New York, 1. Aufl. Auflage, 2006, ISBN 978-3-540-26872-7.
- [PPA⁺13] Pfohl, Prof. Dr. Dr. h. c. Hans Christian, Prof. Dr. Dr. h. c. Hans Christian Pfohl, Prof. Dr. Dr. h. c. Ulli Arnold, Prof. Dr. Klaus Ballarini, Prof. Dr. Ingolf Bamberger, Prof. Dr. Stefan Behringer, Prof. Dr. Christoph J. Börner, Prof. Dr. Ronald Gleich, Prof. Dr. Winfried Hamel, Prof. Dr. Eberhard Hamer, Prof. Dr. habil. Thomas Hering, Stefan Hofmann, Dr. Detlef Keese, Prof. Dr. Jörn Axel Meyer, Prof. Dr. Herfried M. Schneider, Dipl. Kfm. Mike Schulze, Prof. Dr. Hermann Simon, Dr. Dr. Aurelio J. F. Vincenti, Prof. Dr. Thomas Wrona und Prof. Dr. Cornelia Zanger: *Betriebswirtschaftslehre der Mittel- und Kleinbetriebe - Größenspezifische Probleme und Möglichkeiten zu ihrer Lösung*. Erich Schmidt, Berlin, 5. Aufl. Auflage, 2013, ISBN 978-3-503-15469-2.
- [SM13] Schumacher, Jörg und Matthias Meyer: *Customer Relationship Management strukturiert dargestellt - Prozesse, Systeme, Technologien*. Springer-Verlag, Berlin Heidelberg New York, 2013, ISBN 978-3-642-17107-9.
- [TW15] Thome, Rainer und Axel Winkelmann: *Grundzüge der Wirtschaftsinformatik - Organisation und Informationsverarbeitung*. Springer-Verlag, Berlin Heidelberg New York, 2015, ISBN 978-3-662-46732-9.
- [Web12] Weber, Rainer: *Technologie von Unternehmenssoftware - Mit SAP-Beispielen*. Springer-Verlag, Berlin Heidelberg New York, 12. Aufl. Auflage, 2012, ISBN 978-3-642-24423-0.
- [Wik16] Wikipedia: *Texterkennung* — *Wikipedia, The Free Encyclopedia*, 2016. <https://de.wikipedia.org/wiki/Texterkennung>.

Glossar

F | G | K | M | O | P | R | S | T | W

F

Fat-Client In einer Client-Server-Umgebung ist der Fat-Client für die anwendungsspezifische Durchführung von Applikationsaufgaben und der Präsentation zuständig.

G

Geschäftsdaten Komplexe Datenstrukturen, welche persistent in einer Datenbank gespeichert werden.¹⁴²

K

Kernprozess Unternehmensprozesse, die den Unterschied zur direkten Konkurrenz ausmachen und zur Wertschöpfung des Unternehmens beitragen. Sie sind die für das Unternehmen entscheidenden, Wettbewerbsvorteil ausmachenden Geschäftsprozesse, die von einem ERP-System unterstützt werden müssen.¹⁴³

M

Merkmalskatalog Beschreibt häufig vorkommende gemeinsame Merkmale von KMU.¹⁴⁴

O

Optical Character Recognition Bezeichnet die automatisierte Texterkennung innerhalb von Bildern.¹⁴⁵

P

Prozessdefinition Die Repräsentation eines Geschäftsprozesses in einer Form, welche die Modellierung und Ausführung durch ein Workflow-Management-System unterstützt. Prozessdefinitionen bestehen aus einem Netzwerk von Aktivitäten (Aufgaben) und deren Beziehungen, Kriterien zur Identifikation von Start- und Endergebnissen sowie Informationen über die individuellen Aktivitäten (Aufgaben) wie beispielsweise zugeordnete Teilnehmer, IT-Anwendungen, Daten, etc.¹⁴⁶

R

Regelkreis Der Regelkreis ist ein Vorgang, bei dem eine zu regelnde Größe mit einer anderen Größe (der Führungsgröße) verglichen und abhängig vom Ergebnis dieses Vergleiches im Sinne einer Anpassung an die Führungsgröße beeinflusst wird.¹⁴⁷

¹⁴² [Web12, S. 14]

¹⁴³ [Lei12, vgl. S. 49 ff.]

¹⁴⁴ [Lei12, vgl. S. 49]

¹⁴⁵ [Wik16]

¹⁴⁶ vgl. [Coa94, S. 11]

¹⁴⁷ vgl. <http://www.michael-broetje.de/index.htm>

S

Supply-Chain-Management-System Schaffen eine unternehmensübergreifende Informationstransparenz über Bedarfe, Kapazitäten und Bestände der Unternehmen, so dass zum einen der Aufbau einer Entscheidungsunterstützung betrieblicher Abläufe in Echtzeit gefördert wird, zum anderen die komplexe Betrachtung von Szenarien, die mehrere Unternehmen umfassen, im Rahmen von Planungsprozessen für ein kooperatives Prozesscontrolling und Exception Handling erreicht werden kann.¹⁴⁸

Systemkomponente Ein Software, welche über spezifizierte Schnittstellen mit einem Workflow-Management-System interagiert

T

Thin-Client Im Gegensatz zum Fat-Client ist der Thin-Client lediglich für die Präsentation zuständig. Die Durchführung von Applikationsaufgaben wird vom Server übernommen.

W

WAPI-Funktion Menge von häufig benötigten API-Aufrufen und Austauschformaten, die in ihrer Summe benötigt werden, um die fünf Schnittstellen (engl. Interfaces) aus dem WfMC-Referenzmodell (siehe 2.5 auf Seite 21) zu unterstützen.¹⁴⁹

Web-App Eine Anwendung, die ihre Inhalte über den Browser aus dem Internet holt und nicht auf dem Endgerät installiert werden muss.¹⁵⁰

Workflow Management Siehe ?? ?? Seite ??

Workflow Management Coalition Non-Profit-Organisation bestehend aus verschiedenen Herstellern, Anwendern und Forschungseinrichtungen im Umfeld des Workflow Management

Workflow-Management-System Ist die Software für das Workflow-Management und die Ausführung modellierter Arbeitsabläufe

¹⁴⁸ [Hel15]

¹⁴⁹ vgl. [Coa95, S. 46]

¹⁵⁰ [Dud16]

Erklärung

Hiermit erkläre ich, dass ich meine Arbeit selbstständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die Arbeit noch nicht anderweitig für Prüfungszwecke vorgelegt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Mittweida, 25. Oktober 2016